Pendahuluan

Android merupakan sistem operasi yang perkembangannya sangat masif dan cepat. Saat ini, android sudah identik dengan smartphone. Perkembangan aplikasi di android juga sangat cepat, bahkan tahun lalu saja, tiap bulan ada lebih dari 10 ribu aplikasi ditambahkan untuk android.

Yang membuat android makin moncer adalah sarana pendukungnya yang berkembang pula dengan pesat. Seperti perangkat kerasnya yang juga ikut berkembang dengan cepat. Sehingga keragaman fitur dari sistem operasi, hardware, dan operator, membuat semua orang makin cinta dengan android.

Awalnya, sebenarnya yang hendak dijadikan platform untuk android adalah Netbook. Tapi, kecepatan android yang sangat cepat membuat android (atau bagiannya) terimplementasi juga di televisi, bahkan mobil.

Dengan bertumbuhnya jumlah aplikasi yang berjalan di platform android dengan cepat, peluang untuk para developer android makin luas sehingga proses pembuatan aplikasi menjadi sangat penting dikuasai.

1.1 Kenapa Buku Ini Penting?

Sistem operasi android sangat mudah untuk dipelajari, dan Google pun menyediakan banyak library untuk mengimplementasikan aplikasi yang kaya dengan mudah. Tapi masih ada yang kurang –sebagaimana disinyalir oleh para pengembang android–, yaitu dokumentasi yang jelas termasuk contoh-contohnya. Karena itulah buku ini penting untuk membantu Anda belajar programming dan hacks di android, dari mulai pemula hingga advanced.

Buku ini dilengkapi dengan contoh-contoh program yang membuat Anda bisa melakukan pemrograman secara nyata, yaitu Anda bisa belajar dengan contoh program yang benar-benar bisa melakukan sesuatu atau memberi solusi pada sebuah masalah. Anda tinggal mengedit sedikit contoh program ini agar bisa sesuai dengan masalah Anda. Memang tidak semuanya disertai dengan contoh proyek riil, beberapa hal hanya diberi penjelasan, dan kodenya. Tapi pasti dengan mudah Anda bisa mengimplementasikannya di proyek Anda.

Untuk melengkapi pemahaman Anda tentang buku ini, Anda bisa menggunakan referensi tambahan agar pemahaman Anda sempurna, di antaranya adalah:

- Source code Android: http://source.android.com/
- Halaman developer android: http://developer.android.com/
- Forum developer android: http://www.svcAndroid.com/
- Direktori Open Source: http://osdir.com/
- Thread diskusi Stack Overflow: http://stackoverflow.com/
- Forum developer Talk Android: http://www.talkandroid.com/android-forumss

1.2 Sekilas Tentang Android

Sejarah sistem operasi Android dimulai sejak diumumkannya Open Handset Alliance di akhir tahun 2007. Sebenarnya ide untuk membuat sistem operasi open source buat peranti embedded sudah ada sejak lama. Karena backing dari Google yang sangat agresiflah, maka Android bisa sangat terkenal hanya dalam beberapa tahun saja.

Beberapa perusahaan telekomunikasi, kini mulai memiliki atau mena-warkan perangkat android dalam produk-produknya. Tidak hanya ponsel, tapi perangkat lain seperti tablet, netbook, televisi, bahkan hingga ke peranti embedded di mobil memiliki sistem operasi android.

Gambar 1.1 Logo Android

Sebenarnya android pada awalnya tidak dikembangkan oleh google, tapi dikembangkan oleh sebuah perusahaan bernama Android Inc. Karena google melihat banyaknya user yang online dengan perangkat mobile, maka google mengira bahwa perangkat mobile ini memiliki masa depan yang cerah. Sehingga Android Inc diakusisi oleh Google di tahun 2005.

Pada waktu yang sama, Apple di tahun 2007 juga mengembangkan sebuah sistem operasi iOS untuk iPhone-nya dengan fitur-fitur yang

sangat inovatif, seperti multitouch dan adanya open market untuk aplikasi.

Android pun kemudian meniru fitur ini dan cepat beradaptasi untuk menambahkan fitur ini dan fitur tambahan lainnya, seperti programmer bisa punya kontrol yang lebih terhadap sistem operasi, dan multitasking.

Android punya banyak tambahan yang cocok untuk dunia korporat, seperti exchange support, remote wipe, dan Virtual Private Network (VPN). Ini mirip dengan BlackBery yang dikembangkan oleh RIM.

Dengan beragamnya peranti dan adaptasi, android berkembang pesat dan jumlah user-nya pun bertambah. Tapi bagi developer, ini juga berarti ada tantangan yang potensial.

Karena aplikasi harus bisa adaptasi dengan ukuran layar dan resolusi yang berbeda, ada juga keyboard, sensor hardware, versi OS, konfigurasi sistem, dan kecepatan data wireless yang berbeda-beda.

Tiap variasi dari aspek-aspek tersebut bisa menyebabkan munculnya perilaku yang tidak terprediksi. Namun mengecek satu-satu untuk variasi tersebut juga tidak memungkinkan. Karena jumlahnya yang sangat banyak.

Karena itu, sejak awal Android sudah diatur agar bisa memiliki tampilan yang seragam, dan user experience yang sama di semua platform. Caranya dengan melakukan abstraksi pada perbedaan hardware. Sistem operasi Android menutup aplikasi dari modifikasi yang hardware-specific.

Tapi di sisi lain juga memungkinkan fleksibilitas untuk melakukan tune up pada aspek yang diperlukan.

Google menyediakan banyak kemudahan untuk pengembang software pihak ketiga, seperti plugin Android Development Tool (ADT) untuk

Eclipse (dan juga tool standalone) termasuk di dalamnya kemampuan untuk logging secara realtime, emulator realistis yang menjalankan kode ARM native dan pelaporan error.

Berikut ini beberapa hal penting seputar android:

- Android adalah sistem operasi embedded yang sangat bergantung pada kernel linux untuk layanan-layanan core-nya, tapi Android bukanlah linux embedded. Misalnya Android tidak mendukung utilitas standar linux, seperti X-Windows dan GNU C libraries tidak didukung.
- Penulisan program untuk android applications menggunakan framework java, tapi ini bukanlah java. Karena library standar java seperti Swing tidak didukung. Library lain seperti timer tidak disarankan, karena sudah diganti dengan library default dari android, yang dioptimalkan untuk penggunaan di lingkungan embedded yang terbatas.
- OS android merupakan sistem operasi open source, artinya developer bisa melihat semua source code sistem, termasuk stack radio. Source code ini termasuk salah satu bahan belajar untuk melihat cara kerja android, terutama ketika kekurangan dokumentasi. Tapi android juga punya beberapa software proprietary yang tidak mungkin diakses oleh developer, seperti navigasi GPS.

1.2.1 Peranti Android

Hardware yang didukung android memiliki beberapa fitur yang sama, agar bisa didukung oleh sistem operasi, yaitu:

- Bootloader: Memulai loading dari boot image selama startup.

- Boot image: Kernel dan RAMdisk.
- System image: Sistem operasi android dan aplikasi.
- Data image: Data user yang disimpan selama siklus power.
- Recovery image: File yang digunakan untuk membangun kembali atau meng-update sistem.
- Radio image: File dari radio stack.

File-file di atas disimpan di memory nonvolatil sehingga tetap terproteksi ketika power dari peranti down. Flash memory ini hanya bisa dibaca saja, atau sering disingkat sebagai read-only memory (ROM).

Pada saat startup, mikroprosesor mengeksekusi bootloader untuk meload kernel dan RAMDISK ke RAM agar kecepatan akses menjadi cepat. Mikroprosesor kemudian mengeksekusi instruksi dan page portion dari sistem, dan image data ke RAM.

Radio image ada di baseband prosesor yang melakukan koneksi ke hardware radio, microprocessor unit (MPU), synchronous dynamic random access memory (SDRAM atau RAM), dan flash memory (ROM).

Ukuran layar android dinyatakan dalam pixel. Tapi resolusi dpi-nya (dot per inch) sangat bergantung pada layar fisik dari peranti yang digunakan. Misalnya hanphone tertentu punya layar 3,3 inchi dengan piksel sebanyak 320x480.

Ini sama dengan 180 piksel per inchi. Tapi ini dianggap memiliki densitas layar medium oleh android. Di mana rata-ratanya adalah 160 dpi. Peranti android biasanya juga punya sensor kamera CMOS, Bluetooth (BT), Wi-Fi dan (802.11), walaupun tentu saja bervariasi.

Gambar 1.2 Sebuah ponsel android standar

1.2.2 Tab Android

Ketika Apple memperkenalkan iPad, pabrikan android tertarik untuk memperkenalkan komputer tablet semodel dengan iPad tersebut. Apakah sebenarnya definisi komputer tablet?

Komputer tablet adalah komputer yang punya layar minimal 4,8 inchi dan memiliki konektivitas Wi-Fi. Selain itu, karena tab-tab punya layanan 3G atau CDMA maka juga bisa dianggap sebagai sebuah smartphone.

Komputer tablet android biasanya memiliki hard drive asli untuk data storage. Beberapa merek awal yang muncul, contohnya adalah Dell Streak dan Galaxy Tab dengan ukuran 7 inchi dan 10 inchi.

Gambar 1.3 Contoh sebuah Tab (tablet) android

1.2.3 Peranti Lain

Karena android ini platformnya generik untuk peranti embedded, maka dapat juga diterapkan di peranti lain di luar smartphone dan tablet. Contohnya Android pernah diterapkan di sebuah mobil dengan nama Roewe 350 yang dikembangkan oleh pabrikan Shanghai Automotive Industry.

Gambar 1.4 Mobile ROEWE yang menggunakan Android di dalamnya

Di mobil ini, android digunakan untuk navigasi GPS, dan juga bisa dipakai untuk web browsing.

Android juga dipakai di Google TV, yang merupakan kerja sama antara Google dengan perusahaan lain, di mana Google berperan mengembangkan softwarenya, Sony untuk televisinya, Intel untuk prosesornya, dan Logitech untuk set-top box-nya.

Google TV menggabungkan antara internet dengan televisi, selain itu juga bisa mengakses android market dari televisi.

1.2.4 Perbedaan Hardware di Peranti Android

Hardware di peranti android bisa berbeda-beda tergantung pada variasi yang dimilikinya. Berikut ini beberapa perbedaan atau variasi hardware yang lazim ada di peranti android.

Layar

Ada dua teknologi utama yang dipakai untuk layar, yaitu liquid crystal display (LCD) dan light-emitting diode (LED). Pilihan spesifiknya untuk android adalah thin-film transistor (TFT).

LCD dan active-matrix organic LED displays (AMOLED) merupakan pengembangan dari TFT yang memiliki fitur masa hidup lebih lama. Salah satu keuntungan AMOLED adalah tidak perlu adanya backlighting, sehingga warnanya lebih jelas, dan konsumsi dayanya lebih rendah.

Secara keseluruhan, android memiliki 3 jenis ukuran layar, yaitu kecil, normal, dan besar. Selain itu, resolusi atau kerapatan pikselnya juga bisa dibagi menjadi beberapa, yaitu rendah, medium, dan tinggi.

Sebenarnya kerapatan piksel bervariasi, tapi bisa bervariasi sesuai dengan tabel berikut ini.

Tipe layar	**Densitas rendah (~120 ppi)**	**Densitas medium (~160ppi)**	**Densitas tinggi (~240ppi)**
Layar kecil	QVGA (240x320), 2,6" – 3"		
Layar normal	WQVGA (240x400), 3,2" - 3,5"	HVGA (320x480)	WVGA (450x800) 3,3" – 4 "
Layar besar		WVGA (480x800) 4,8" -5,5 "	

Metode Input

Android juga memiliki perbedaan user input, yang utama adalah touchscreen. Touchscreen memungkinkan user untuk berinteraksi dengan display visual. Ada tiga tipe teknologi layar sentuh alias touch screen ini.

- Resistif: Dua lapis material diletakkan di atas lapisan kaca di layar. Ketika jari atau stylus menekan atau memberi tekanan maka kedua lapisan ini akan bersentuhan dan lokasi dari penekanan bisa ditentukan. Layar resistif ini sangat murah, tapi tidak mendukung multitouch hingga barusan saja sudah bisa mendukung multitouch. Efektivitas model ini sekitar 75%.
- Kapasitif: Material bermuatan diletakkan di atas layar kaca. Ketika tangan Anda atau objek konduktif yang menyentuh layar, maka muatan akan berpindah, dan mengubah nilai kapasitansinya. Ini akan diukur untuk menentukan lokasi penyentuhan. Ini sangat efektif, nilainya hingga 90%. Tapi akurasinya tidak seakurat resistif.

- Surface Acoustic Wave: Model ini menggunakan metode yang akan mengirim dan menerima gelombang ultrasonik. Ketika tangan Anda atau objek lain menyentuh layar maka gelombang akan diserap. Maka gelombang diukur untuk menentukan lokasi penyentuhan. Ini merupakan solusi paling awet, ini lazim diterapkan di ukuran layar besar, misalnya di ATM.

Semua peranti android biasanya menggunakan salah satu metode, entah resistif atau kapasitif. Dan biasanya juga mendukung multitouch.

Selain touch screen, peranti android juga biasanya punya metode alternatif untuk mengakses layar. Berikut ini beberapa metode alternatif lain.

- D-pad (directional pad): Sebuah joystik untuk menunjukkan atas bawah dan kanan kiri.

Gambar 1.5 Dpad di sebuah peranti android

- Trackball: Sebuah bola berputar yang mirip dengan peranti mouse di komputer.

Gambar 1.6 Trackball untuk android

- Trackpad: Sebuah permukaan kotak yang berfungsi sebagai alat penunjuk.

Kebanyakan smartphone punya tiga sensor dasar:

- Accelerometer tiga sumbu: untuk mengatur gravitasi.
- Magnetometer tiga sumbu: untuk mengatur medan magnet ambien.
- Sensor temperatur untuk mengatur temperatur ambien.

1.2.5 Fitur Android

Berikut ini beberapa fitur android yang membuat banyak orang kesengsem dan tertarik dengan android. Dengan memahami fitur ini, Anda bisa memanfaatkan android semaksimal mungkin.

- Multiprocess dan App Widgets

  OS android tidak membatasi prosesor ke satu program saja. Tapi sistem bisa mengatur prioritas aplikasi dan thread dalam satu aplikasi. Ini memungkinkan program background bisa dijalankan ketika user membiarkan peranti berjalan di background,

misalnya ketika sedang main game. Sementara di belakangnya ada program yang tetap berjalan.

App Widget merupakan aplikasi mini yang bisa di-embed di aplikasi lain, seperti Home Screen. App widget bisa memproses event, seperti memulai streaming audio, atau meng-update temperatur luar, sambil aplikasi lain tetap terus berjalan.

Multiprocessing memungkinkan user memperoleh kesenangan yang banyak ketika mengoperasikan android. Namun, ini juga punya kekurangan, yaitu android bisa boros baterai jika banyak aplikasi dijalankan sekaligus.

- Touch, Gesture, dan Multitouch

  Touch screen adalah antarmuka user yang sangat intuitif untuk peranti hand held. Jika piawai memakainya, maka touch screen sangat membantu Anda.

  Ketika jari menyentuh layar, Anda bisa men-drag atau melemparkan jari Anda untuk mengoperasikan layar. Multitouch memberikan cara untuk berinteraksi dengan touch screen di lebih dari satu tangan pada saat yang bersamaan. Misalnya untuk zoom atau rotasi.

  Beberapa event touch nantinya akan dijelaskan di bagian event-event user interface.

- Keyboard Hard dan Soft

  Peranti android juga memiliki keyboard, keyboard ini ada 2 jenis, yaitu jenis hard/fisik dan soft yang berupa software. Keyboard fisik biasanya lebih cepat untuk mengetik.

Tapi, beberapa gadget hanya menyertakan keyboard soft saja dengan berbagai macam alasan, utamanya adalah agar lebih praktis, nyaman, dan desainnya ergonomis.

Salah satu kekurangan keyboard soft adalah memakan tempat. Karena perlu bagian untuk memperlihatkan keyboard tersebut.

1.2.6 Agar Program Kompatibel

OS Android selalu diperbarui secara periodik, karena itu agar program Anda kompatibilitasnya tinggi, Anda perlu memerhatikan beberapa saran-saran berikut:

- Jangan gunakan API internal atau API yang tidak didukung.
- Jangan langsung memanipulasi setting tanpa meminta persetujuan dari user. Rilis masa depan bisa jadi membatasi setting untuk alasan keamanan. Misalnya dulu aplikasi bisa mengaktifkan GPS atau data roaming sendiri, tapi sekarang tidak bisa lagi dilakukan.
- Jangan membuat layout terlalu kompleks. Karena kalau terlalu kompleks bisa membuat crash.
- Jangan mengasumsikan hardware akan selalu mendukung aplikasi. Tidak semua peranti android memiliki hardware yang dibutuhkan, karena itu cek dulu apakah peranti android Anda memiliki hardware yang dibutuhkan oleh program atau tidak?
- Pastikan orientasi device tidak mengganggu aplikasi atau menjadikan aplikasi menjadi error. Misalnya orientasi layar.
- Gunakan library android sebelum menggunakan ilbrary Java. Library android ini dibuat khusus untuk peranti embedded dan

memiliki banyak persyaratan untuk aplikasi. Untuk kasus lain, library java juga diperlukan. Jika keduanya bisa dipakai, usahakan untuk menggunakan library android.

- Hati-hati dengan alokasi memory, pembuatan variabel, dan penggunaan objek. Usahakan untuk me-reuse objek dan hindari garbage collection. Alokasi memory bisa di-tracking menggunakan Dalvik Debug Monitor.

1.3 Instalasi Android Emulator

Dalam membuat program, Anda harus instal terlebih dahulu Android Emulator. Di mana android emulator ini membutuhkan JDK, Eclipse, dan lainnya. Ada beberapa tombol shortcut untuk memakai android emulator ini nantinya, berikut ini beberapa di antaranya.

Tombol Shortcut	**Fungsinya**
Escape	Back button
Home	Home button
F2	PageUp Menu button
Shift-F2	PageDown Start button
F3	Call/Dial button
F4	Hangup/EndCall button
F5	Search button
F7	Power button
Ctrl-F3, Ctrl-KEYPAD_5	Camera button

Ctrl-F5, KEYPAD_PLUS	Volume up button
Ctrl-F6, KEYPAD_MINUS	Volume down button
KEYPAD_5	DPAD center
KEYPAD_4, KEYPAD_6	DPAD left, DPAD right
KEYPAD_8, KEYPAD_2	DPAD up, DPAD down
F8	Toggle cell network on/off
F9	Toggle code profiling (saat -trace set)
Alt-ENTER	Toggle fullscreen mode
Ctrl-T	Toggle trackball mode
Ctrl-F11, KEYPAD_7 Ctrl-F12, KEYPAD_9	Memutar orientasi layar ke layout sebelumnya atau setelahnya.

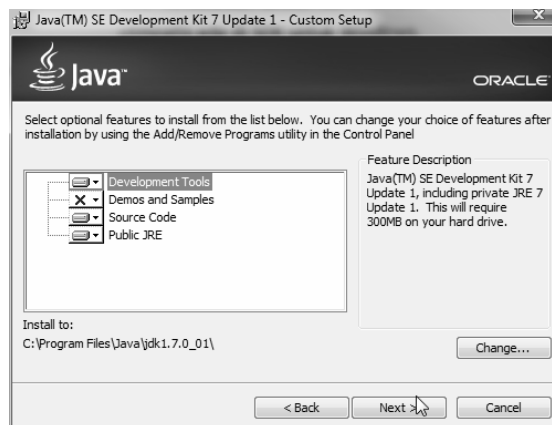
Pada umumnya, pengetesan pertama dilakukan menggunakan ponsel android. Ini memungkinkan fungsionalitas penuh, dan bisa mengetahui permasalahan apa yang muncul secara real time, di mana kadang tidak bisa ditampilkan di emulator.

Jika Anda ingin menggunakan peranti android sebagai platform pengembangan, maka Anda bisa menancapkan peranti android Anda ke USB menggunakan kabel USB yang ada di ponsel. Dan pastikan driver USB terdeteksi. Driver ini otomatis mendeteksi karena driver secara otomatis ada di SDK untuk Windows.

Pertama instal dulu JDK yang bisa diambil dari http://www.oracle.com/technetwork/java/javase/downloads/index.html. Setelah di-download, JDK perlu dipasang/diinstal di komputer dengan langkah seperti berikut:

1. Jalankan installer Java SDK dengan klik dua kali pada installer.

Gambar 1.7 Installer Java SDK

2. Kemudian set komponen Development tools, source code, dan JRE diaktifkan.

Gambar 1.8 Penentuan komponen-komponen yang hendak dijalankan

3. Klik **Next** dan tunggu hingga penyalinan file installer selesai.

Gambar 1.9 Penyalinan file installer dari Java SDK

4. Di **Install To**, Anda dapat menentukan lokasi instalasi.

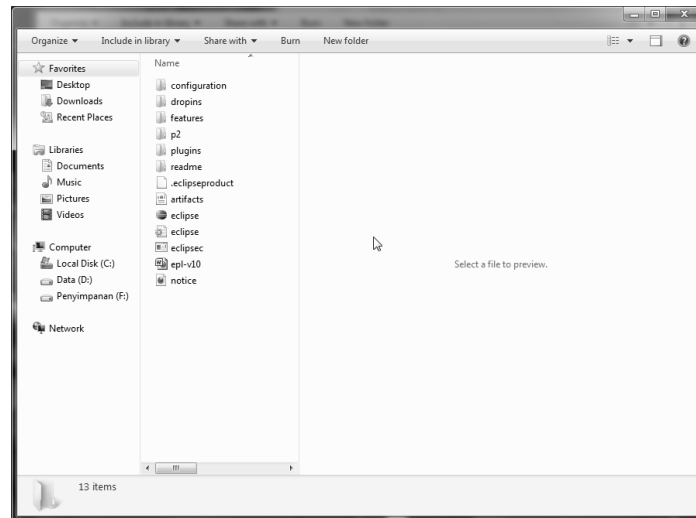
Gambar 1.10 Lokasi instalasi di Install to

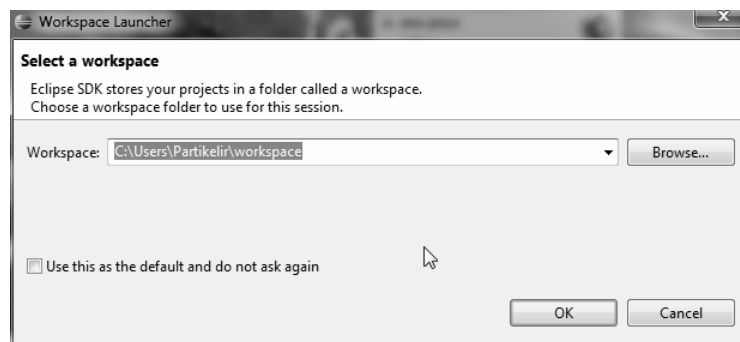
5. Tunggu hingga instalasi ini selesai.

Gambar 1.11 Instalasi sedang berlangsung

6. Anda bisa klik **Finish** di jendela terakhir yang menyatakan JDK sukses diinstal.

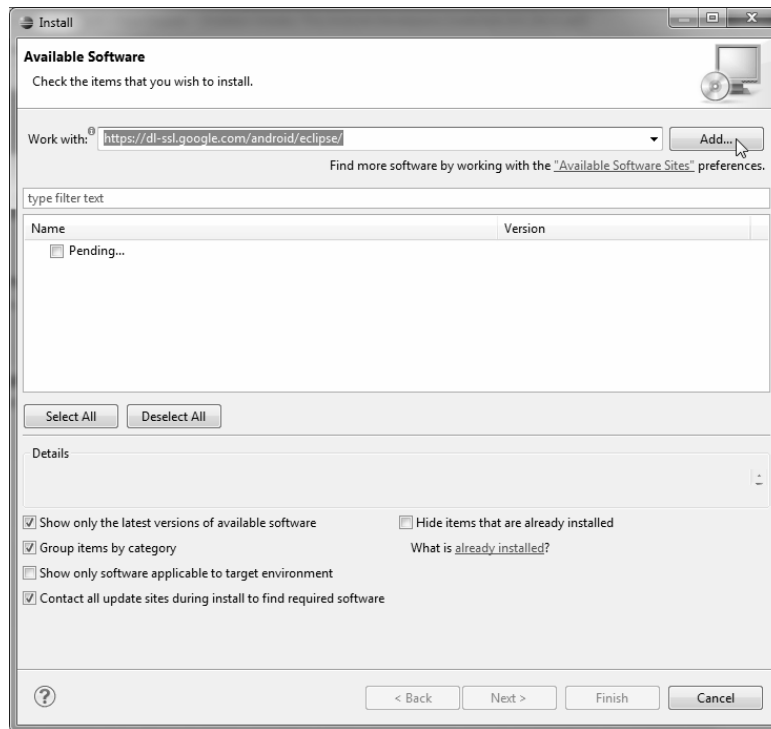
Gambar 1.12 JDK sukses diinstal

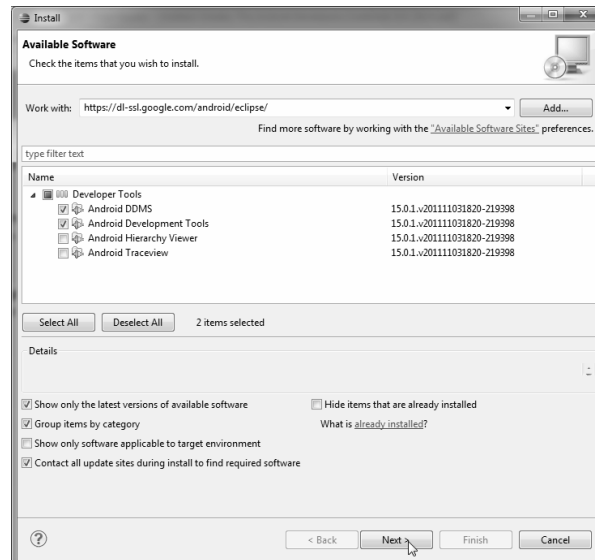
Setelah JDK selesai diinstal, berikutnya adalah menginstal eclipse. Eclipse sendiri cara memasangnya hanya berupa mengekstrak file arsip dari installer-nya saja. Hasil ekstraksinya terlihat seperti berikut ini.

Gambar 1.13 Hasil ekstraksi dari file

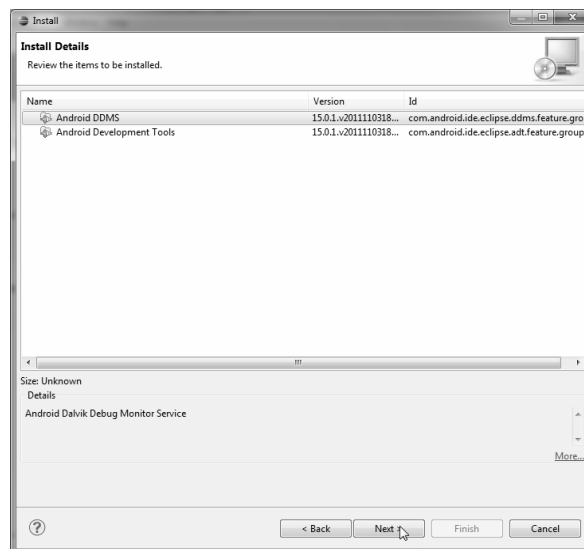
Ketika pertama kali dijalankan, eclipse memilih workspace di **Workspace**. Klik **OK**. Cek pada **Use this as the default and do not ask again**.

Gambar 1.14 Klik Workspace

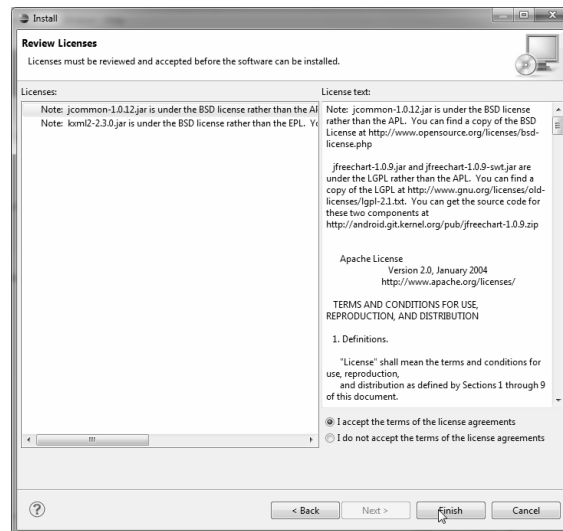
Berikutnya konfigurasikan Eclipse dengan cara seperti berikut:

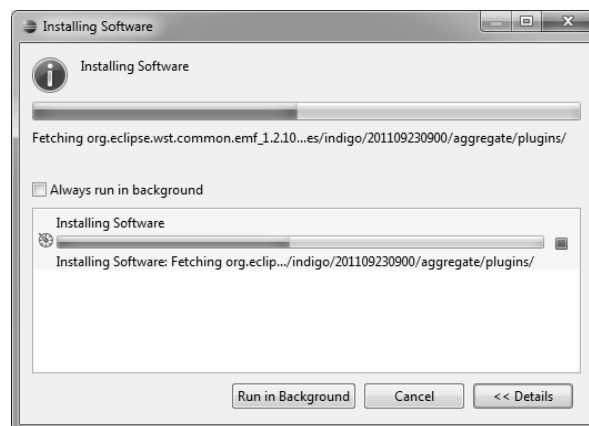
1. Klik **Help > Install New softwares**.

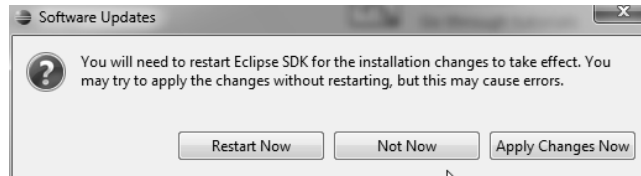
2. Buka alamat https://dl-ssl.google.com/android/eclipse/ di **Available software**. Klik **Add** untuk melihat daftar file yang bisa di-download.

Gambar 1.15 Memperlihatkan Available software

3. Berikutnya muncul nama developer tool untuk android dari url sebelumnya. Cek pada **Android DDMS** dan **Android Development Tools**. Kemudian klik lagi **Next**.

Gambar 1.16 Cek pada Android DDMS dan Android Development Tools

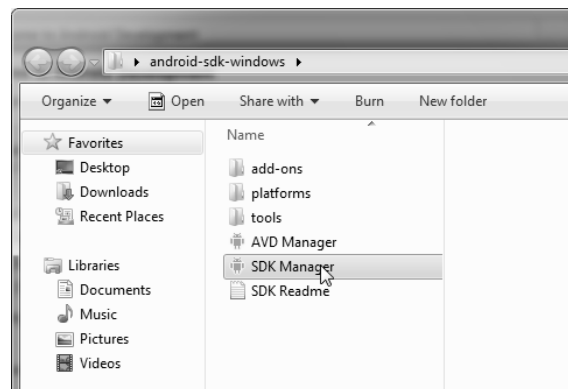
4. Di **Install Details** terlihat detail informasi program yang akan diinstal ke Eclipse. Klik **Next** untuk menuju ke tahap selanjutnya.

Gambar 1.17 Install Details

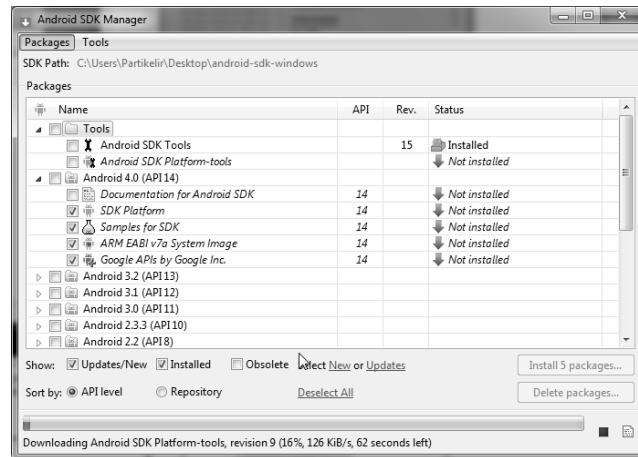
5. Cek pada **I accept the terms of the license agreement**. Kemudian klik **Finish**.

Gambar 1.18 I accept the terms of the license agreements

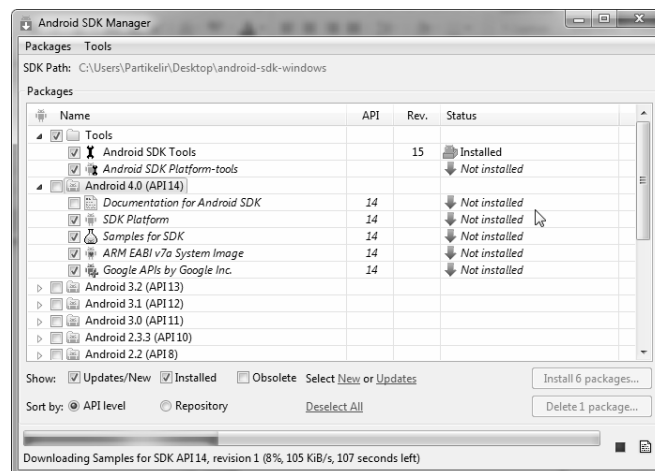
6. Tunggu saat proses instalasi sedang berlangsung. Kalau mau dijalankan di background, klik pada **Run in background**.

Gambar 1.19 Instalasi software tambahan ke dalam eclipse

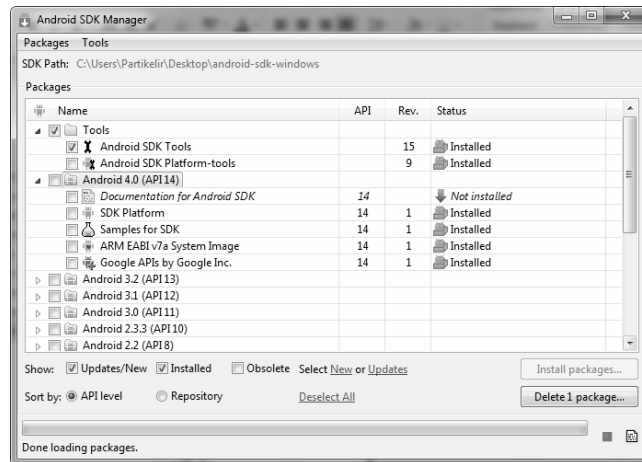
7. Kalau muncul perintah restart, klik **Restart Now** untuk me-restart komputer.

Gambar 1.20 Restart now untuk me-restart komputer

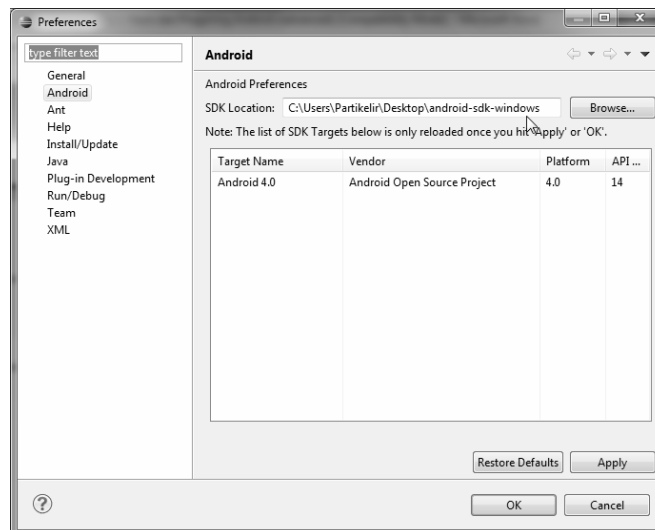
Setelah JDK dan eclipse dijalankan, berikutnya Anda tinggal menginstal Android SDK. Cara menginstalnya cuma dengan mengekstrak dan kemudian mengikuti langkah berikut:

1. Buka folder hasil ekstraksi, kemudian klik pada **SDK Manager**.

Gambar 1.21 Klik pada SDK Manager

2. Cek pada versi android yang Anda inginkan untuk diinstal, tapi apa pun versinya, Android SDK tools dan platform tools juga harus diinstal.

Gambar 1.22 Pengecekan bagian yang sudah terinstal dan yang belum

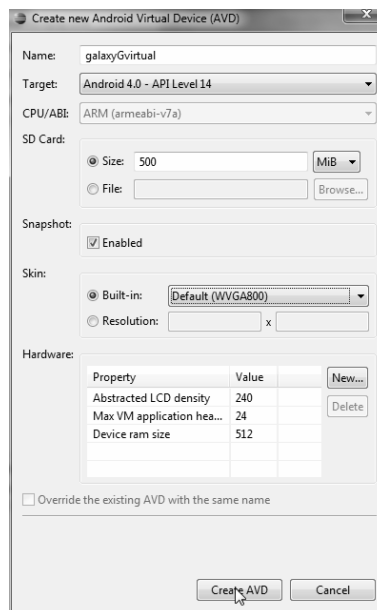
3. Cek pada bagian-bagian yang ingin diinstal, kemudian klik pada **Install packages**.

Gambar 1.23 Pengecekan install packages

4. Kalau sudah, maka tulisan **Not installed** menjadi **Installed**.

Gambar 1.24 Not installed menjadi installed

5. Buka Eclipse lalu klik **Windows > Preferences** di **Eclipse**.
6. Tentukan lokasi sdk di **SDK Location**. Ini untuk mengintegrasikan antara SDK android dengan eclipse.

Gambar 1.25 Pengisian lokasi SDK android

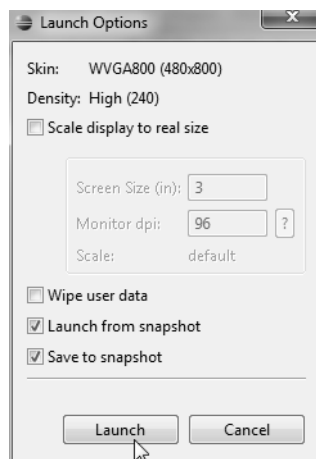
7. Kemudian buka **Windows > AVD Manager**. Ini akan membuka AVD manager. Klik **New** untuk membuat AVD (android virtual device) baru.

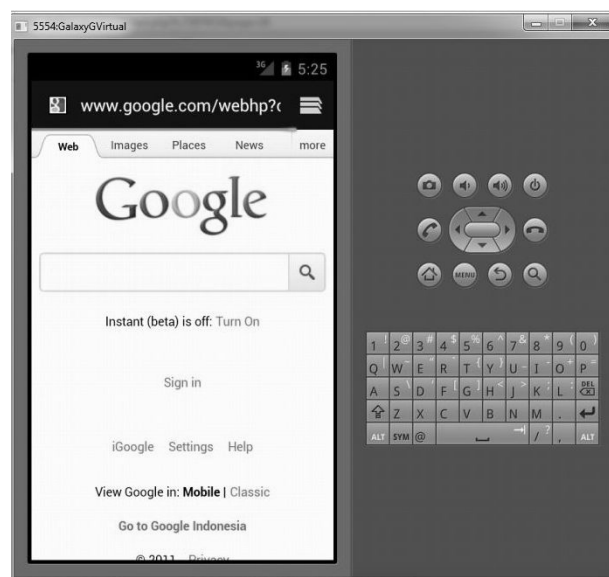
Gambar 1.26 AVD Manager dibuka

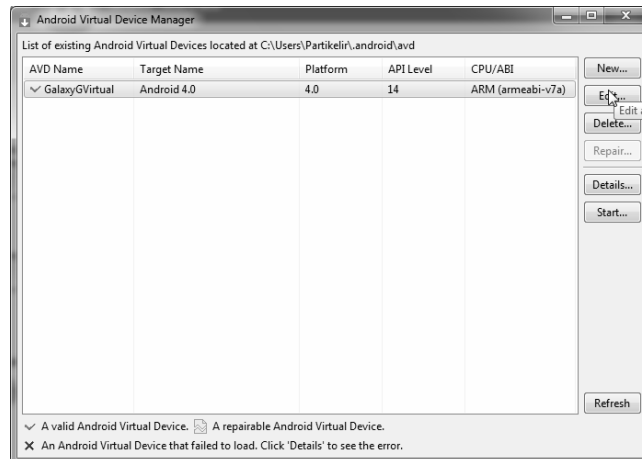
8. Isikan nama AVD di **Name**, isikan target platform di **Target**. Tentukan ukuran kartu di **SD card**.

Gambar 1.27 Penentuan target

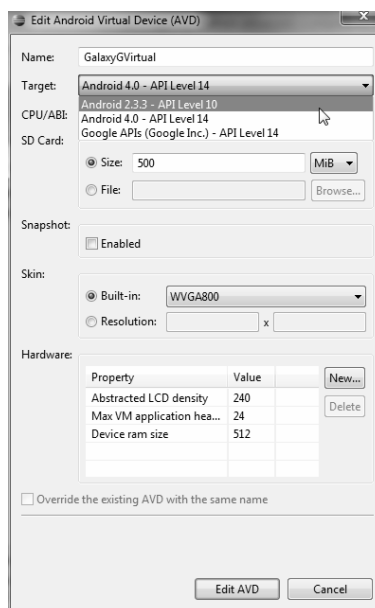
9. Klik **Create AVD** untuk membuat avd ini.

Gambar 1.28 AVD sudah terbuat

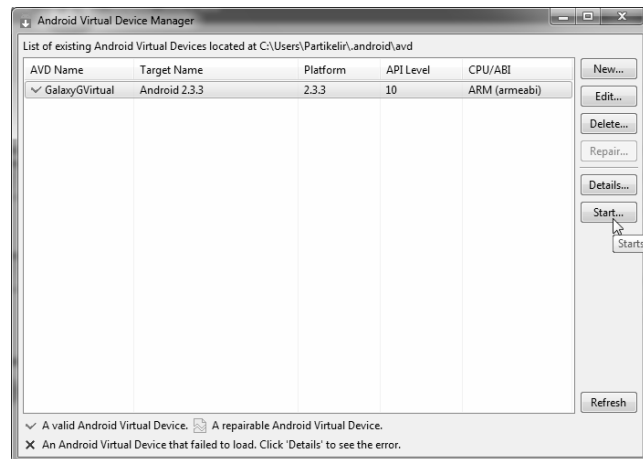
10. Kalau mau di-launch, klik pada AVD yang Anda inginkan, kemudian kilk **Start**.

Gambar 1.29 Pembuatan AVD

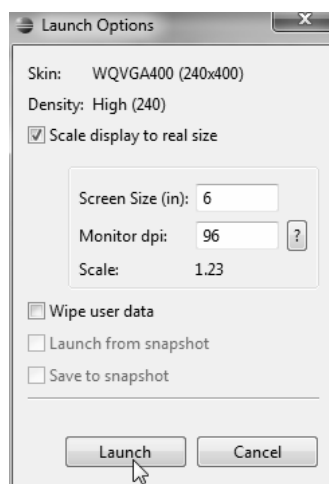
11. Emulator akan menjalankan window di komputer yang dikembangkan, yang bentuknya mirip dengan ponsel android. Dan berjalan di atas instruksi ARM. Startup awal ini akan sangat lambat, bahkan di komputer yang canggih sekalipun. Walaupun ada teknik untuk mengonfigurasi emulator agar bisa meniru atau mengemulasi beberapa aspek dari peranti android asli, seperti telepon yang masuk, pembatasan data rate, perubahan orientasi, tapi beberapa fitur seperti sensori dan audio/video tidak sama, karena itu emulator ini hanya digunakan untuk memvalidasi fungsi-fungsi dasar saja.

Gambar 1.30 Emulator sudah dijalankan

12. Di satu AVD, Anda bisa mengedit/mengubah versi AVD dengan klik pada AVD tersebut kemudian klik **Edit**.

Gambar 1.31 Klik pada AVD kemudian klik pada Edit

13. Kemudian ubah yang mau Anda edit, misalnya pengubahan target dari satu versi Android ke versi sebelumnya.

Gambar 1.32 Pengeditan versi android

14. Klik **Edit AVD** untuk menerapkan perubahan ini.

15. Kemudian klik pada AVD ini dan klik **Start**.

Gambar 1.33 Klik pada Android Virtual Device Manager

16. Tentukan beragam opsi di **Launch options**, dan klik **Launch**.

Gambar 1.34 Pengaturan beragam opsi sebelum launch

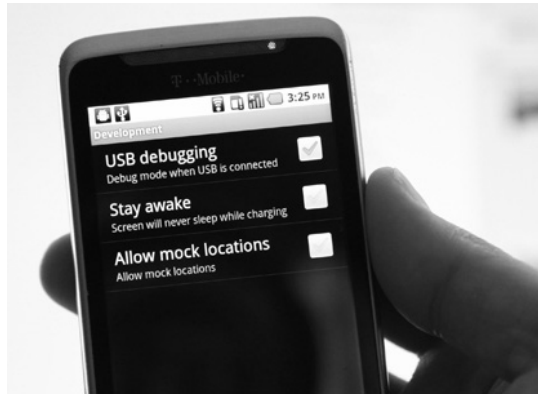
17. Hasilnya, android akan muncul tapi dengan target yang berbeda. Dengan cara ini, Anda bisa mengecek program Anda dengan versi yang berbeda-beda untuk menjamin kompatibilitasnya.

Gambar 1.35 AVD muncul dengan versi android yang berbeda

1.3.1 Hack: Menggunakan Peranti Android Asli untuk Debugging

Jika Anda melakukan development android di Windows dan ingin langsung mengoneksikan ke peranti android betulan dalam debugging-nya, maka Anda tinggal menginstal usb driver sesuai dengan peranti Anda.

Nantinya, kalau sudah terinstal dan dikenali oleh Windows, maka ketika Anda menjalankan program di Eclipse secara langsung, maka Anda bisa memilih mana peranti yang Anda ingin jalankan. Pilih ke peranti android Anda.

Pertama set perangkat Anda untuk bisa menginstal aplikasi non-market. Caranya di peranti Anda, klik **Settings > Applications > Development** di peranti Anda.

Gambar 1.36 Memberi tanda cek pada opsi USB debugging

Kemudian aktifkan USB Debugging. Di Android 4.0, setting-nya ada di opsi **Settings > Developer**.

Berikut ini beberapa URL untuk memperoleh driver untuk merek HP Anda. Jika HP Anda merek lokal/Cina seperti Nexian, IMO atau Cross, maka cek dulu merek aslinya apa di Cina. Setelah itu baru bisa dicari atau di-browse via internet.

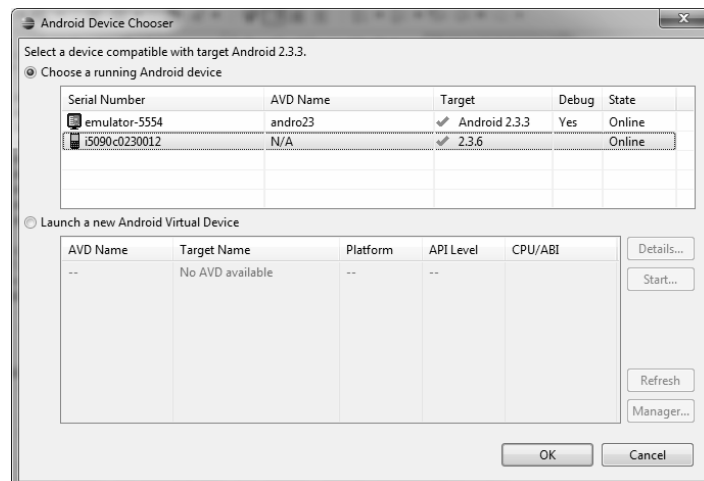
- Acer: http://www.acer.com/worldwide/support/mobile.html
- Alcatel one touch: http://www.alcatel-mobilephones.com/global/Android-Downloads
- Asus: http://support.asus.com/download/
- Dell: http://support.dell.com/support/downloads/index.aspx?c=us&cs=19&l=en&s=dhs&~ck=anavml
- Foxconn: http://drivers.cmcs.com.tw/
- Fujitsu:http://www.fmworld.net/product/phone/sp/android/develop/

- Fujitsu Toshiba: http://www.fmworld.net/product/phone/sp/android/develop/
- Garmin-Asus: https://www.garminasus.com/en_US/support/pcsync/
- HTC: http://www.htc.com
- Huawei: http://www.huaweidevice.com/worldwide/downloadCenter.do?method=list&flay=software&directoryId=20&treeId=0
- KT Tech: http://www.kttech.co.kr/cscenter/download05.asp
- Kyocera: http://www.kyocera-wireless.com/support/phone_drivers.htm
- Lenovo: http://developer.lenovomm.com/developer/download.jsp
- LGE: http://www.lg.com/us/mobile-phones/mobile-support/mobile-lg-mobile-phone-support.jsp
- Motorola: http://developer.motorola.com/docstools/USB_Drivers/
- Pantech: http://www.isky.co.kr/cs/software/software.sky?fromUrl=index
- Pegatron: http://www.pegatroncorp.com/download/New_Duke_PC_Driver_0705.zip
- Samsung: http://www.samsung.com/us/support/downloads
- Sharp: http://k-tai.sharp.co.jp/support/

- SK Telesys: http://www.sk-w.com/service/wDownload/wDownload.jsp
- Sony Ericsson: http://developer.sonyericsson.com/wportal/devworld/search-downloads/driver?cc=gb&lc=en
- ZTE: http://support.zte.com.cn/support/news/NewsDetail.aspx?newsId=1000442

Ketika sudah mendapatkan driver-nya, Anda bisa menginstal driver tersebut dengan cara seperti berikut:

1. Hubungkan peranti android Anda ke port usb komputer.
2. Klik kanan di **Computer** dari desktop atau **Windows Explorer**, kemudian klik **Manage**.
3. Pilih **Device** di panel kiri.
4. Cari dan expand device lain di panel kanan.
5. Klik kanan pada nama peranti (contohnya Nexus Tab) kemudian pilih **Update Driver Software**. Ini akan menjalankan **Hardware Update Wizard**.
6. Pilih **Browse my computer** untuk driver software dan klik **Next**.
7. Klik **Browse** dan cari lokasi folder driver USB.
8. Klik **Next** untuk menginstal driver.

Kalau Driver sudah terinstal, ketika Anda me-run proyek Anda, maka muncul **Android Device Choose** di Eclipse, di mana Anda bisa memilih peranti android fisik untuk me-run program Anda.

Gambar 1.37 Pilihan untuk me-run program di peranti Android berupa ponsel

1.4 Dasar Pembuatan Aplikasi di Android

Sebuah aplikasi android memiliki beberapa fungsi, misalnya untuk mengedit tulisan, memainkan file audio, membunyikan alarm, atau membuka kontak telepon.

Fungsi-fungsi ini bisa diklasifikasikan menjadi empat komponen yang berbeda, yang ditentukan oleh base class java yang berbeda.

Fungsionalitas	**Java base class**	**Contohnya**
Memfokuskan pada hal yang dilakukan user	Activity	Mengedit note, memainkan game.
Proses background	Service	Memainkan audio, update ikon weather.

Menerima pesan	BroadcastReceiver	Memicu alarm ketika ada permintaan.
Simpan dan ambil data	ContentProvider	Membuka daftar kontak di telepon.

- **Activity**

  Merupakan hal yang ada pada aplikasi Android. Sebuah activity, biasanya berupa satu layar dalam aplikasi. Activity ini dinyatakan dengan sebuah class tunggal yang merupakan perpanjangan (*extends*) dari class Activity.

  Class akan menampilkan interface yang terdiri atas Views dan respons untuk menjalankan aksi tertentu. Dan kebanyakan aplikasi Android terdiri atas beberapa layar. Misal, ketika kita akan mengirim sms maka akan terlihat satu layar yang berisi daftar kontak, kemudian layar kedua untuk menulis isi sms.

  Kemudian layar ketiga yang menampilkan status sms, kemudian layar keempat yang berisi daftar sms yang lama, kemudian layar lainnya bisa jadi berisi setting sms, edit profil, dan sebagainya.

- **Intent Receiver**

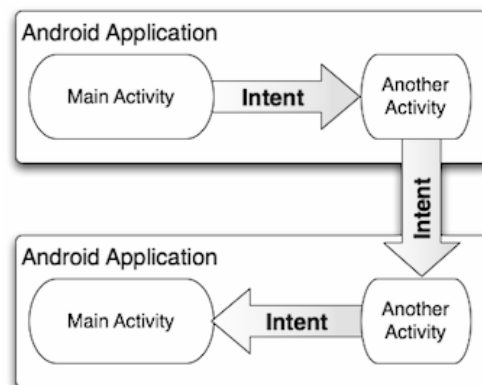
  Android menggunakan class Intent untuk melakukan aksi, berubah dari layar satu ke layar yang lain. Dua hal utama yang perlu diperhatikan pada struktur class Intent adalah aksi yang dilakukan dan data yang dibawa.

  Aksi yang dilakukan dinyatakan dengan perintah seperti VIEW, PICK, EDIT, dan sebagainya. Sedangkan datanya dinyatakan pada URI (Uniform Resource Indicator). Misal, Anda ingin menampilkan halaman web tertentu, maka dibuat class Intent dengan

aksi VIEW dan alamat situs dimasukkan ke dalam URI. Contoh perintahnya terlihat seperti berikut ini.

```
New Intent(android.content.Intent.VIEW_ACTION,
ContentURI.create("http://elexmedia.com"));
```

Sebuah class intent tidak dapat melakukan sendiri, namun membutuhkan sebuah intent filter dan intent receiver. Intent receiver akan menerima semua activity yang diberikan, kemudian oleh intent filter akan dipilih aksi mana yang akan dijalankan.

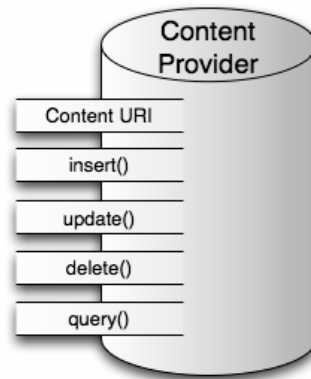
Misal, ketika ada intent ACTION_WEB_SEARCH maka yang dijalankan adalah dalam format http dan bukan format ftp atau yang lainnya. Singkatnya, intent ini digunakan jika Anda ingin berpindah dari satu activity ke activity lainnya.

Gambar 1.38 Komponen-komponen aplikasi android

- **Service**, merupakan aplikasi Android yang berjalan tanpa UI, dan berada di balik layar. Misalnya, sebuah service MP3 Player maka mp3 akan tetap dimainkan walaupun pengguna menjalankan activity lainnya seperti browsing, chatting.

Service ini dapat dijalankan secara manual atau secara otomatis ketika Android pertama kali dihidupkan. Service dapat dimatikan secara manual oleh user atau mati sampai tidak diperlukan.

Player akan menjalankan service menggunakan **Context.startService()** sehingga akan berjalan di belakang layar dan audio akan tetap hidup sampai dimatikan secara manual/ otomatis.

Gambar 1.39 Service di android

- **Content Providers**, merupakan sebuah class yang berisi method-method yang memperbolehkan aplikasi lain menyimpan dan mengambil data yang di-handle oleh sebuah content provider. Content Provider ini berguna ketika kita merancang aplikasi yang dibuat dapat di-share untuk aplikasi lainnya.

Gambar 1.40 Content provider

Tiap aplikasi terbuat dari satu komponen di atas atau lebih. Keempat komponen tersebut dibuat instance-nya oleh sistem operasi Android sesuai kebutuhan. Aplikasi lain juga diperkenankan untuk menggunakannya, tapi dengan permission tertentu.

Tiap komponen punya siklus dibuat – fokus – defokus – dihancurkan. Kecuali ContentProvider, semua komponen diaktivasi oleh pesan asinkronus yang disebut *Intent*.

Intent ini berisi sekelompok informasi yang menjelaskan komponen. Ini menyediakan metode untuk melewatkan informasi antar-komponen. Karena aktivitas selalu membutuhkan interaksi dengan user, sebuah jendela otomatis akan dibuat untuk tiap aktivasi.

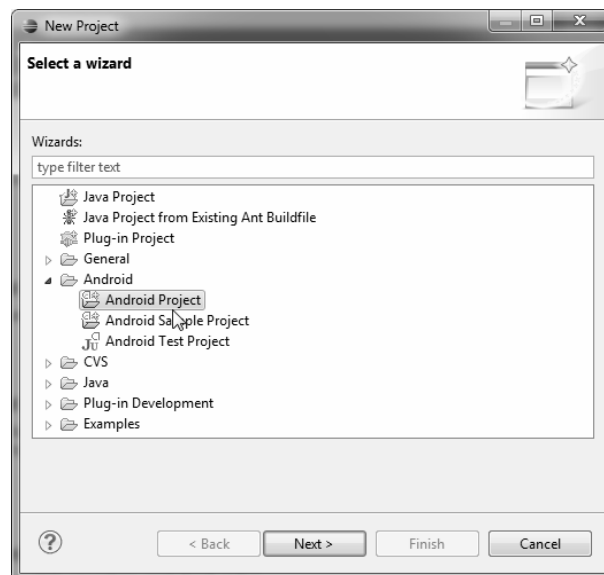
Semua aktivitas akan memperluas/extend class abstrak Activity atau salah satu sub class-nya. Kunci untuk tiap activity adalah method onCreate() yang gunanya untuk memulai/inisialisasi user interface, membuat button listener, memulai parameter, dan memulai thread.

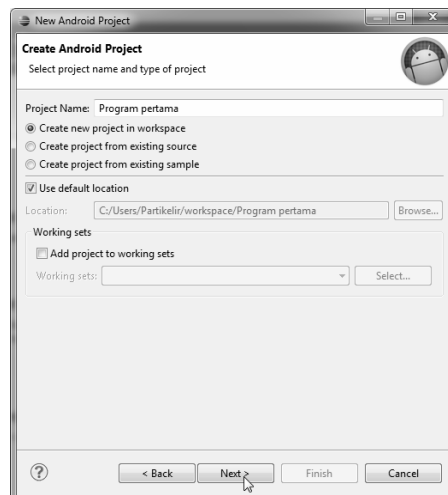
Jika main activity tidak diciptakan oleh proyek maka activity perlu untuk ditambahkan. Cara menambahkan activity adalah:

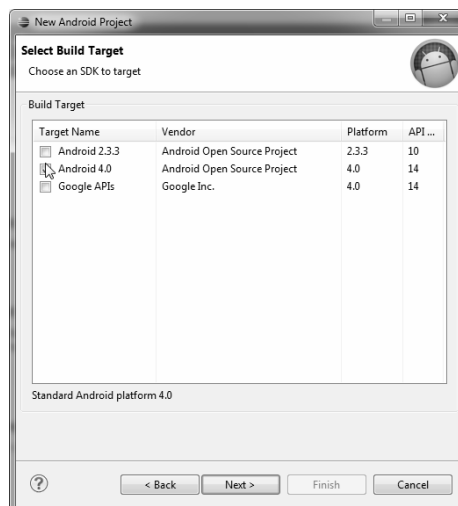
1. Buat class untuk meng-extend Activity. Di Eclipse, caranya dengan klik kanan pada proyek, kemudian klik **New** → **Class**, dan menentukan android.app.Activity sebagai super class.
2. Lakukan override fungsi onCreate(). Di Eclipse, caranya adalah dengan klik kanan pada file Class, kemudian memilih **Source** -> **Override/Implement Methods** dan mencek pada method onCreate.

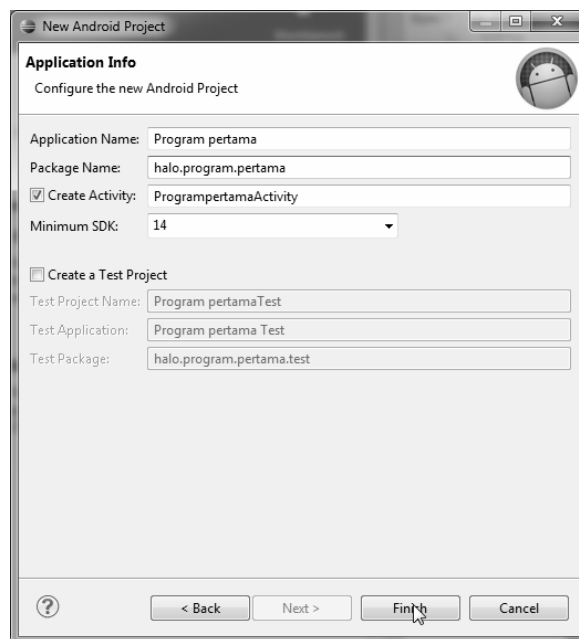
1.4.1 Membuat Proyek Pertama Kali

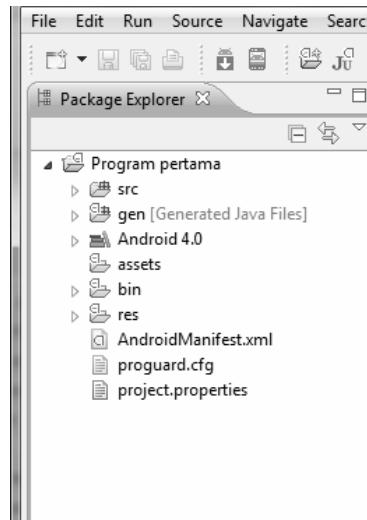
Untuk lebih jelas bagaimana membuat program pertama kali, ikuti langkah-langkah berikut:

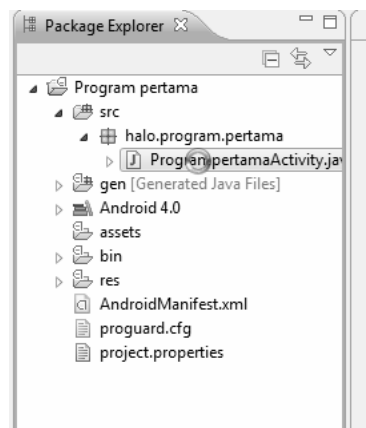
1. Buka Eclipse, buat proyek pertama kali dengan klik **File > New > Project**.
2. Pilih proyek **Android > Android project**.

Gambar 1.41 Pemilihan jenis proyek Android > Android project

3. Muncul wizard **New Android Project**. Pilih pada **Create new project in workspace**, dan cek pada **Use default location**.

Gambar 1.42 Wizard, new android project

4. Berikutnya, klik pada versi android yang Anda inginkan di-build target. Klik lagi **Next**.

Gambar 1.43 Pemilihan build target

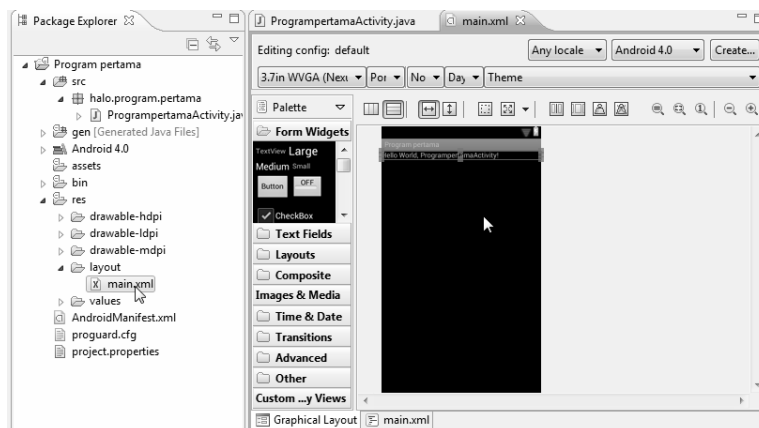
5. Isikan nama-nama aplikasi (app name) di **Application name**.
6. Isikan package name di **Package name**.
7. Cek pada **create activity** untuk membuat activity.

Gambar 1.44 Pengisian info aplikasi

8. Klik **Finish**, maka Eclipse secara otomatis akan membentuk sebuah folder baru berisi berbagai file pendukung yang digunakan untuk membuat aplikasi baru. Anda bisa melihatnya di **Package explorer**.

Gambar 1.45 Package explorer untuk melihat/meng-explore file-file di dokumen

9. Di **src**, Anda bisa melihat nama package dan di dalam nama package, Anda bisa mengklik pada file berekstensi .java untuk melihat source code-nya.

Gambar 1.46 Kode dari file package

10. Kodenya seperti berikut, kode ini berarti bahwa jika activity dibuat, maka ditampilkan tampilan di R.layout.main:

```
package halo.program.pertama;

import android.app.Activity;
import android.os.Bundle;

public class ProgrampertamaActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

11. Untuk melihat tampilan preview dari layout, Anda bisa membuka file /res/layout/main.xml. Kalau file ini dipilih, muncul preview dari program Anda di tab main.xml seperti gambar berikut.

Gambar 1.47 Main.xml memperlihatkan layout program

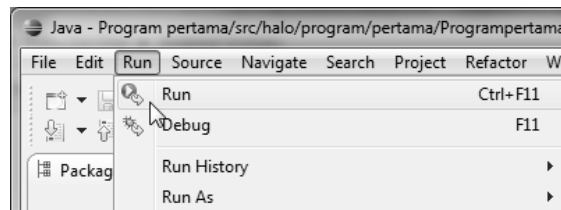
12. Adapun kode dari main.xml ini aslinya adalah:

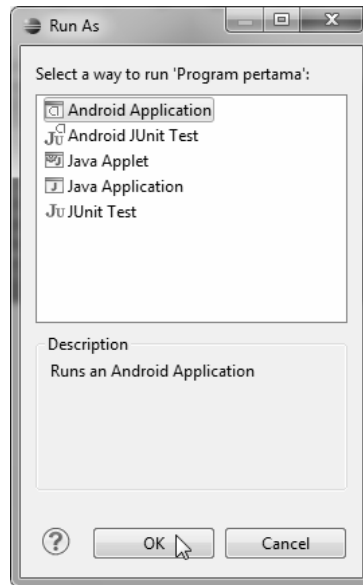
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
android:orientation="vertical"
android:layout_width="fill_parent"
```

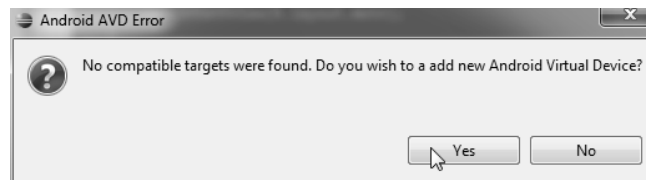
```
android:layout_height="fill_parent"
>
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/hello"
/>
</LinearLayout>
```

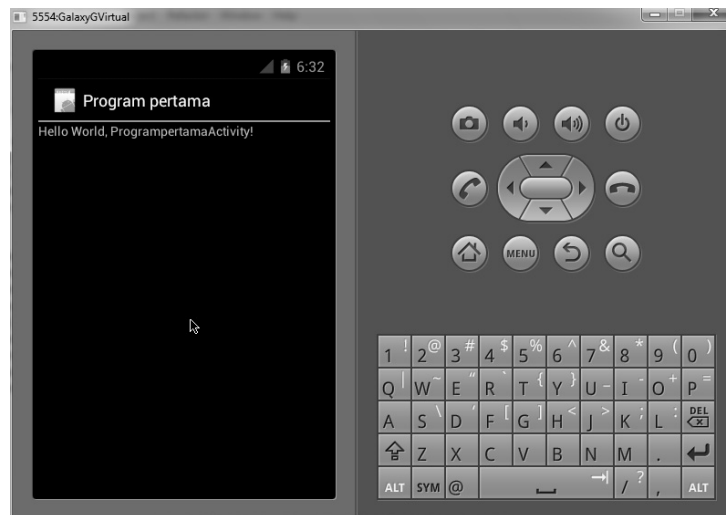
13. Seperti pada kode di atas, android:layout_width="fill_parent" maksudnya objek akan mengisi penuh layar atau kontainer.

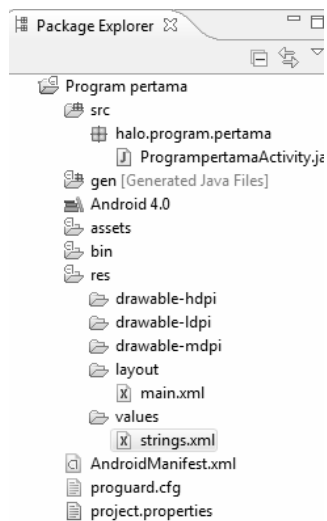
14. Pada baris: android:layout_height="wrap_content" maksudnya objek akan mengisi layer sesuai dengan karakter yang ada.

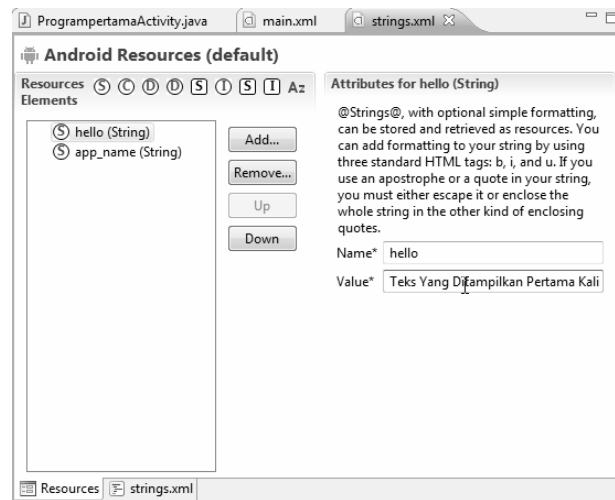
15. Dalam kondisi standar, tampilan string yang ada adalah string default, yaitu @string/hello. Jalankan saja dahulu untuk melihat bagaimana tampilannya di emulator dengan klik pada **Run > Run** atau klik **CTRL + F11**.

Gambar 1.48 Menu untuk Run

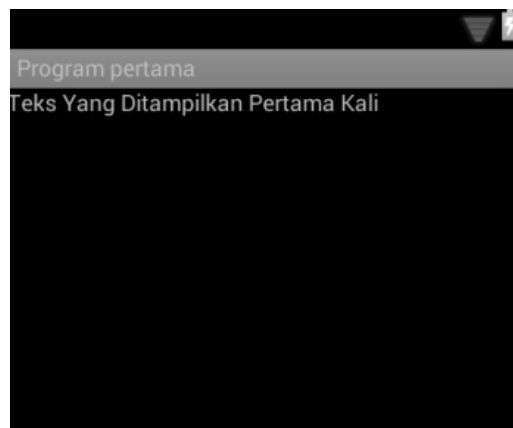
16. Jalankan sebagai Android Application di **Select a way to run + nama_program**.

Gambar 1.49 Pemilihan Run As Android Application

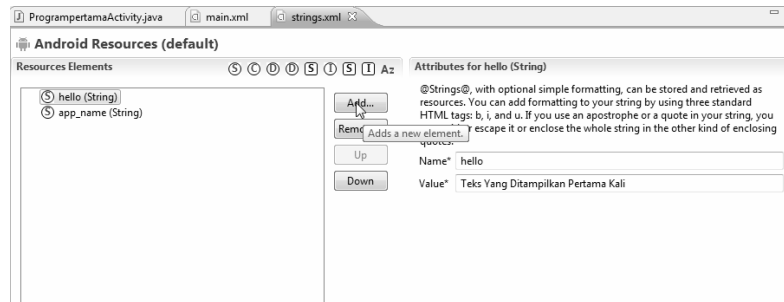
17. Jika belum ada AVD yang kompatibel dengan versi android yang Anda jalankan, muncul pemberitahuan **Android AVD Error**. Klik **Yes**.

Gambar 1.50 Android AVD error seandainya belum ada AVD yang punya versi yang sama

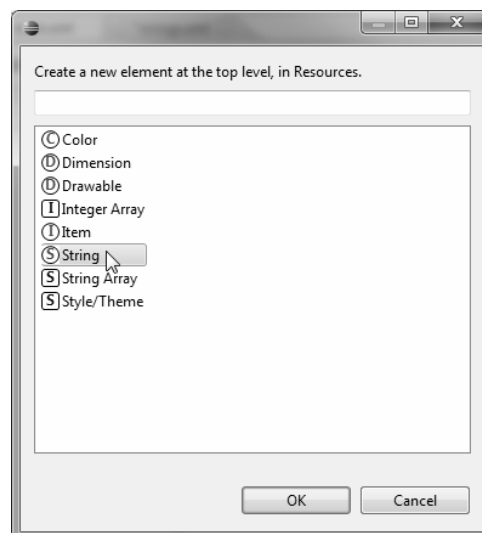
18. Maka tampilan program Anda di AVD terlihat seperti berikut ini.

Gambar 1.51 Tampilan program di AVD

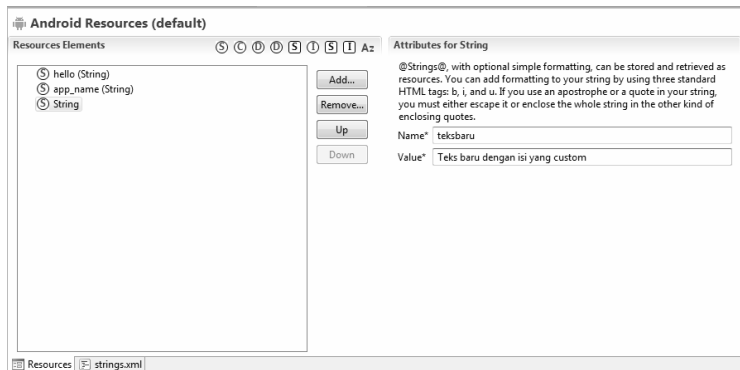
19. Nilai teks yang ditampilkan bisa diubah, kembali ke eclipse, buka res/values/strings.xml.

Gambar 1.52 Membuka /res/values/strings.xml

20. Anda bisa mengubah nilai di **Value**.

Gambar 1.53 Pengubahan teks di Value

21. Kalau teks diedit, tampilan preview berubah menjadi sama.

Gambar 1.54 Tampilan teks sudah berubah

22. Anda bisa menambahkan string lain ke resource dengan klik pada tombol **Add**.

Gambar 1.55 Klik Add untuk menambahkan string baru

23. Untuk elemen, pilih **String**.

Gambar 1.56 Pemilihan string di jenis new element

24. Isikan nama di **Name**, kemudian set isi di **value**.

Gambar 1.57 Pengisian name dan value

25. Tekan **CTRL + S** untuk menyimpan.

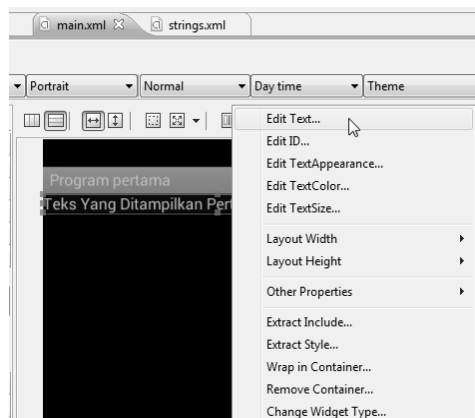
26. Maka nama elemen tersimpan di **Resources elements**.

Gambar 1.58 Pengisian Resources elements

27. Maka kalau file xml-nya dilihat, tampilan menjadi seperti berikut.

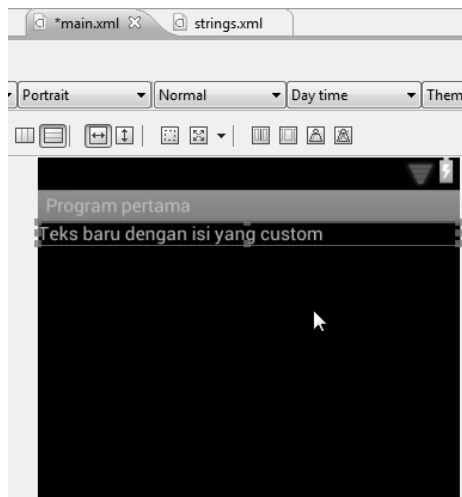
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="hello">Teks Yang Ditampilkan Pertama
Kali </string>
    <string name="app_name">Program pertama</string>
    <string name="teksbaru">Teks baru dengan isi yang
custom</string>

</resources>
```

28. Teks yang sudah dimasukkan akan berubah di file main.xml di /res/layout.

29. Anda bisa mengubah teks langsung dari /res/layout.main.xml dengan klik kanan kemudian pilih menu **Edit text**.

Gambar 1.59 Menu Edit text untuk mengedit teks

30. Kemudian pilih resource string yang barusan dibuat.

Gambar 1.60 Pemilihan resource string yang barusan dibuat

31. Hasilnya, kotak berisi teks yang bersumber dari resource default akan berubah dengan isi teks custom buatan Anda.

Gambar 1.61 Tampilan ketika teks sudah berubah

32. Dari tutorial ini, Anda sudah bisa melakukan pembuatan proyek pertama kali, menjalankannya di AVD, dan pengeditan sederhana pada isi teks.

Ketika membuat program, eclipse menghasillkan berbagai file yang di-generate oleh user, termasuk:

- **src/**: berisi paket java yang ditulis oleh developer atau diimpor untuk aplikasi. Tiap package bisa memiliki file java yang merepresentasikan class yang berbeda-beda.
- **res/layout/**: berisi file XML yang menentukan layout tiap layar.
- **res/values/**: berisi file XML yang digunakan sebagai reference oleh file lain.
- **res/drawable-hdpi/**, **res/drawable-mdpi/**, dan **res/drawable-ldpi/**: merupakan direktori-direktori yang berisi gambar yang

digunakan untuk aplikasi. Punya gambar dengan resolusi high, medium, dan low.

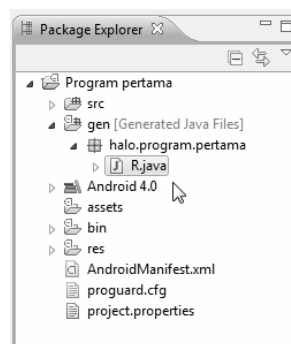
- **assets/**: berisi file-file non media yang digunakan di aplikasi.
- **AndroidManifest.xml**: berisi spek dari proyek di Android.

Selain itu, ada pula file yang di-generate secara otomatis, antara lain:

- **gen/**: berisi kode yang di-generate secara otoamtis. Termasuk di dalamnya adalah class R.java.
- **default.properties**: berisi setting proyek, walaupun ini di-generate secara otomatis, tetap harus dikontrol revisinya.

Resource aplikasi kebanyakan diterapkan dalam bentuk XML, misalnya untuk mendeskripsikan layout. File XML juga digunakan untuk menyimpan nilai seperti string, label dari elemen UI, dan file pendukung lain seperti gambar dan suara.

Ketika saat kompilasi, referensi terhadap resource biasanya disimpan dalam bungkus/wrapper yang di-generate secara otomatis yang disebut R.java.

Proses generate aplikasi ini lazim dilakukan oleh Android Asset Packaging Tool (aapt).

Gambar 1.62 File R.java yang di-generate secara otomatis di folder gen

Berikut ini isi file R.java.

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found.  It
 * should not be modified by hand.
 */

package halo.program.pertama;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
        public static final int teksbaru=0x7f040002;
    }
}
```

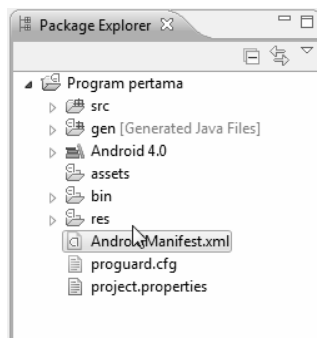
Di sini, tiap resource dipetakan dengan nilai integer tertentu yang unik. Dengan cara seperti ini, class R.java menyediakan referensi ke resource eksternal di dalam kode java.

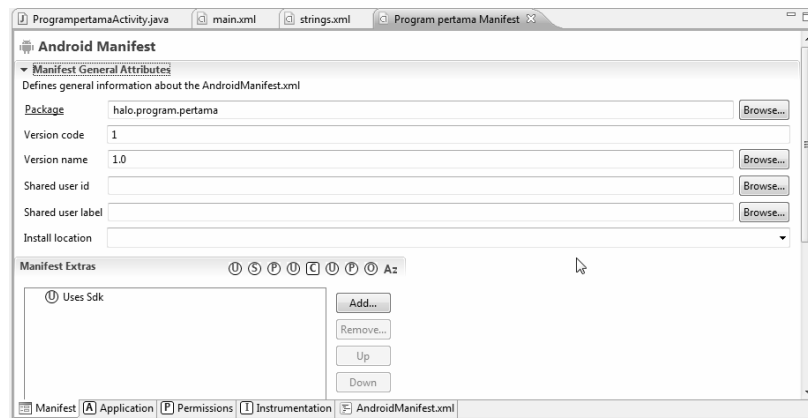
Misalnya untuk referensi ke main R.layout.main, nilai integer digunakan. Untuk reference ke file XML, yang dipakai adalah string "@layout/main". Contohnya untuk mendefinisikan button baru dengan ID bernama home_button, maka tanda plus ditambahkan ke string @+id/home_button. Berikut ini perbedaan pereferensian resource di java dan xml.

Resource	**Reference di Java**	**Reference di XML**
Res /layout/main.xml	R.layout.main	@layout/main
Res /drawable-hdpi/icon.png	R.drawable.icon	@drawable/icon
@+id/home_button	R.id.home.button	@id/home_button
<string name="hello">	R.string.hello	@string/hello

Proyek android sering juga disebut package android karena file-file dipaket dalam satu proyek. Paket Android yang berbeda, bisa memiliki satu nama java package yang sama. Adapun nama package android harus unik di semua aplikasi yang terinstal di peranti android.

Agar Android bisa mengakses program, tiap aplikasi harus mendeklarasi-kan komponennya dalam file Android manifest yang berbasis XML.

Sebagai tambahan, file ini memerlukan permission dan behavior untuk menjalankan aplikasi.

Gambar 1.63 File android manifest berbasis XML

Kalau dibuka, manifest ini bisa memiliki antarmuka visual seperti berikut.

Gambar 1.64 Antarmuka visual dari Manifest

Kalau dibuka file xml-nya, terlihat tampilan visual di atas, dinyatakan dalam kode XML.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="halo.program.pertama"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="14" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".ProgrampertamaActivity" >
            <intent-filter >
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Baris pertama dari kode diperlukan dan merupakan standar untuk semua file xml di android. Gunanya untuk menentukan encoding. Elemen manifest mendefinisikan nama paket android dan versinya.

versionCode merupakan nilai integer yang bisa dievaluasi di program untuk menentukan apakah program nanti upgrade atau downgrade dari versi sebelumnya/sesudahnya.

versionName merepresentasikan format yang bisa dibaca oleh manusia yang bisa direvisi secara mayor atau minor. Elemen aplikasi ini mendefinisikan ikon dan label yang bisa dilihat user dari menu android.

Label adalah sebuah string yang harus cukup pendek untuk ditampilkan di ikon pada peranti user. Nama ini bisa dua kata yang terdiri atas 10 karakter.

Elemen activity mendefinisikan aktivitas utama yang di-launch ketika aplikasi dijalankan. Dan nama ditampilkan di titlebar ketika aktivitas sedang aktif. Nama paket java perlu ditentukan.

Biasanya nama paket java mirip dengan nama paket android. Tapi harus diingat bahwa paket android dan paket java sangat berbeda. Elemen filter inten memberitahukan sistem android mengenai kemampuan komponen.

Bisa digunakan untuk melakukan action ganda, kategori, atau elemen data guna tujuan ini. Elemen SDK mendefinisikan API yang digunakan untuk menjalankan aplikasi. Secara umum, level API dinyatakan seperti berikut:

```
<uses-sdk android:minSdkVersion="integer"
android:targetSdkVersion="integer"
android:maxSdkVersion="integer" />
```

Karena OS android didesain agar bisa forward compatible, maka maxSDK version tidak disarankan. Dan mendefinisikan targetSdkVersion tidaklah diwajibkan. Tapi memungkinkan peranti dengan versi SDK yang sama untuk men-disable setting kompatibilitas yang bisa mempercepat operasi.

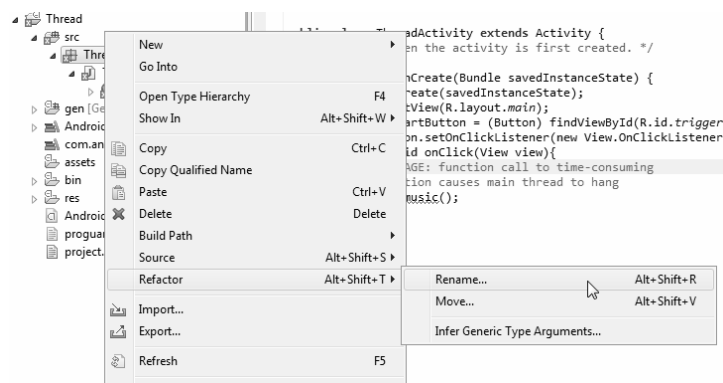
Di sisi lain, minSdkVersion harus selalu ditentukan. Ini untuk memastikan bahwa aplikasi tidak crash ketika dijalankan di platform yang tidak mendukung fitur yang dibutuhkan di aplikasi. Usahakan untuk memilih API level terendah ketika menentukan ini.

AndroidManifest juga punya setting permission yang dibutuhkan untuk menjalankan aplikasi ini.

1.4.2 Rename Bagian dari Aplikasi

Kadang bagian dari proyek android perlu di-rename. Contohnya ketika ada file yang disalin secara manual ke dalam proyek, maka bisa jadi nama aplikasi akan berubah selama pemrograman berlangsung.

Ada tool otomatis yang bisa memastikan referensi silang ter-update secara otomatis. Berikut ini cara untuk me-rename bagian dari aplikasi Anda:

1. Klik kanan pada project kemudian klik **Refactor** > **Move** untuk memindahkan ke folder lain.
2. Klik kanan pada project **Refactor** > **Rename** untuk me-rename proyek ini.

Gambar 1.65 Refactor > rename untuk mengganti pada sebuah package

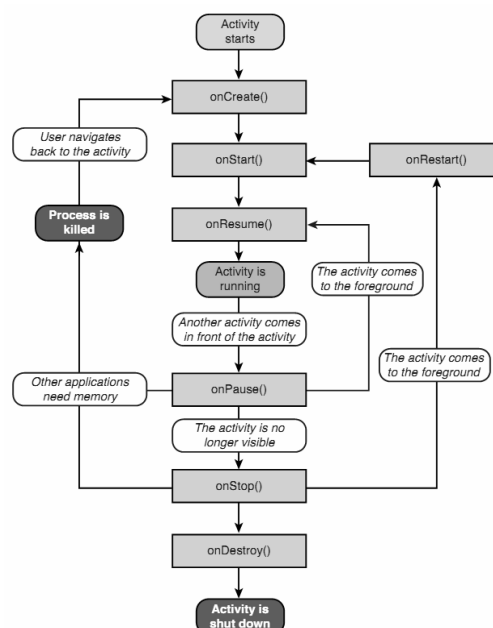
Adapun cara me-rename paket android Anda adalah:

1. Klik kanan pada paket kemudian klik **Refactor > Rename**.
2. Edit AndroidManifest.xml sehingga nama paket baru akan terbuat.

Untuk me-rename class Android seperti komponen: Activity, service, broadcast receiver, caranya seperti berikut:

1. Klik kanan pada file .java dan klik **Refactor** > **Rename** untuk mengganti nama class.
2. Edit AndroidManifest.xml untuk memastikan android:name punya nama komponen baru. Nama file xml lain biasanya harus diedit secara manual untuk mengubah referensi.

1.4.3 Activity dan Siklusnya

Tiap acivity dalam aplikasi selalu punya siklus hidup sendiri-sendiri. Activity ini hanya di-create sekali saja, yaitu ketika fungsi onCreate(). Jika aplikasi keluar, maka fungsi onDestroy() dipanggil. Di antaranya, activity bisa mengalami banyak hal.

Gambar 1.66 Siklus activity

Berikut ini beberapa hal berkaitan dengan activity:

- Mengubah orientasi layar akan menghancurkan dan membuat ulang activity dari dasar.
- Mengklik home button akan membuat activity ter-pause tapi tidak dihancurkan.
- Mengklik ikon aplikasi akan memulai instance baru dari activity, bahkan ketika activity lama belum dihancurkan.
- Membiarkan layar dalam posisi sleep akan mem-pause activity, di mana nanti akan dijalankan ketika layar dikembalikan dari posisi resume. Ini mirip dengan saat menerima telepon masuk.

Hack: Memaksakan Single Task

Ketika aplikasi keluar dari fokus, kemudian di-launch lagi, maka aplikasi bisa punya banyak instance aplikasi di peranti. Akhirnya muncul banyak instance dari activity yang bisa membuat instance berlebih yang memakan banyak memory. Akhirnya beban peranti Android menjadi besar.

Untuk menghindari hal ini, developer bisa mengontrol perilaku tiap aktivitas di Android manifest. Untuk memastikan hanya satu instance aktivitas saja yang berjalan di peranti android Anda, maka Anda bisa memaksakan dengan kode seperti ini.

```
android:launchMode="singleInstance"
```

Kode di atas, membuat hanya ada satu instance di tiap activity di task pada semua keadaan. Semua activity anakan akan di-launch sebagai task tersendiri. Untuk membatasi single task untuk semua activity, gunakan kode berikut:

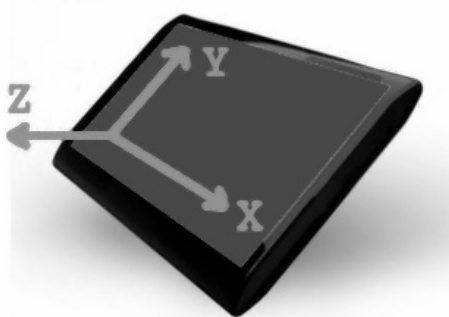
```
android:launchMode="singleTask"
```

Ini memungkinkan activity untuk men-share informasi secara mudah menggunakan task yang sama.

Anda juga bisa mempertahankan kondisi task tertentu, tidak peduli bagaimana user menavigasi activity. Misalnya jika user meninggalkan aplikasi dan melakukan launching lagi, maka biasanya task akan dikembalikan ke kondisi awal. Untuk memastikan user selalu mengembalikan task ke kondisi akhir, maka Anda bisa menggunakan kode berikut:

```
android:alwaysRetainTaskState="true"
```

Hack: Memaksakan Orientasi Layar

Sebuah peranti android yang dilengkapi dengan accelerometer bisa menentukan arah bawah. Ketika device dipindahkan dari posisi portrait ke landscape, maka accelerometer akan melakukan action berupa merotasi view aplikasi sesuai orientasi layar.

Gambar 1.67 Accelerometer yang membantu menentukan posisi peranti android

Activity biasanya dihancurkan dan di-restart tiap kali orientasi layar berubah. Ketika ini terjadi, maka state.kondisi activity bisa hilang, sehingga membuat user menjadi tidak nyaman.

Satu opsi hack untuk meng-handel perubahan orientasi layar secara perlahan-lahan adalah dengan menyimpan kondisi informasi sebelum perubahan, kemudian mengembalikan informasi ini setelah perubahan.

Anda juga bisa memaksa orientasi layar untuk tetap konstan. Untuk tiap activity di Android Manifest, orientasi layar bisa ditentukan di screenOrientation. Misalnya memaksa agar aplikasi dijalankan di mode portrait, Anda bisa menggunakan kode berikut:

```
android:screenOrientation="portrait"
```

Anda juga dapat mengatur mode landscape dengan cara seperti berikut:

```
android:screenOrientation="landscape"
```

Tapi ini juga bisa menyebabkan activity ter-restart ketika keyboard fisik sedang terklik. Sehingga metode ketiga adalah memberitahukan sistem android bahwa aplikasi akan meng-handel orientasi dan keyboard slide out. Caranya dengan kode seperti berikut:

```
android:configChanges="orientation|keyboardHidden"
```

Ini bisa dipakai sendirian, atau dikombinasikan dengan atribut screenOrientation untuk menentukan perilaku yang dibutuhkan ke aplikasi.

Hack: Menyimpan dan Me-Restore Informasi Activity

Ketika activity hendak dimatikan, fungsi onSaveInstanceState() dipanggil. Anda bisa menggunakan ini untuk menyimpan informasi yang relevan.

Ketika activity di-create ulang, maka fungsi onRestoreInstanceState() dipanggil. Fungsinya untuk mengambil informasi yang sudah disimpan sebelumnya.

Ini memungkinkan user tetap memiliki tampilan yang konsisten ketika ada perubahan activity. Tapi kondisi user interface umumnya tidak perlu diatur karena diset secara default oleh sistem.

Fungsi ini berbeda dengan onPause() karena jika ada komponen lain di-launch di depan activity, maka baru fungsi onPause() dipanggil. Ketika Android hendak me-reclaim resource, baru dipanggil lagi fungsi onSaveInstanceState() sebelum activity dimatikan.

Berikut ini contoh kode untuk menyimpan dan me-restore kondisi/state dari instance activity. Kodenya berupa string (teks) dan bilangan floating number.

```
float[] arrrayFLoat = {3.14f, 2.718f, 0.577f};
String namaUser = "Toni";
@Override
protected void onSaveInstanceState(Bundle outState) {
super.onSaveInstanceState(outState);

//menyimpan informasi
outState.putString("name", namaUser);
outState.putFloatArray("array", arrrayFLoat);
}
@Override
public void onRestoreInstanceState(Bundle savedInstanceState)
{

super.onRestoreInstanceState(savedInstanceState);
//mengambil informasi ketika instance di-restore.
namaUser = savedInstanceState.getString("name");
arrrayFLoat = savedInstanceState.getFloatArray("array");
}
```

onCreate() juga berisi savedInstanceState. Yaitu ketika activity meng-inisialisasi ulang setelah sebelumnya dimatikan. onSaveInstanceState() juga dilewatkan ke onCreate(). Pada semua kasus, paket yang tersimpan dilewatkan ke fungsi onRestoreInstanceState(), sehingga wajar jika digunakan untuk me-restore kondisi.

Sebuah program yang sederhana sekalipun, biasanya punya lebih dari satu fungsi. Misalnya sebuah game sederhana yang punya 2 activity,

layar high score untuk mencatat nilai, dan layar game untuk menampilkan game.

Contoh lain adalah notepad yang bisa digunakan untuk melihat view notes, lalu membaca note, dan mengedit note yang terpilih, atau note baru.

Activity utama yang ditulis di AndroidManifest XML dijalankan ketika aplikasi mulai dijalankan. Activity ini bisa me-launch aktivitas lain, biasanya setelah even dipicu. Ini menyebabkan activity untuk mem-pause sementara yang kedua aktif.

Ketika activity kedua berakhir, maka yang utama dibawa ke foreground dan dijalankan. Android menggunakan intent yang implisit semakimal mungkin dan menyediakan framework yang cukup andal untuk fungsionalitas modular.

Multi Activity

Dalam praktiknya, sebuah aplikasi umumnya jarang hanya punya satu activity. Biasanya ada lebih dari satu activity, alias multi-activity. Ketika ada multi activity, Anda menggunakan intent dan multi file layout.

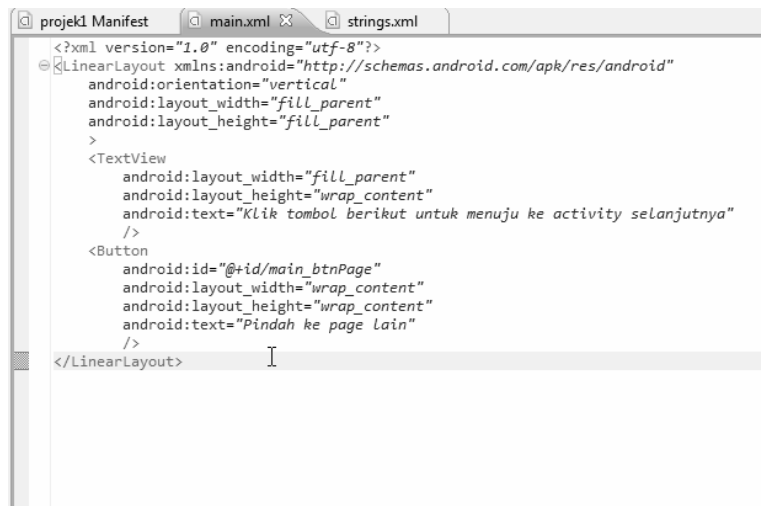
Berikut ini contoh pembuatan sebuah aplikasi yang punya multiactivity. Kalau di bahasa pemrograman visual basic, activity ini identik dengan event. Ikutilah cara-cara berikut ini:

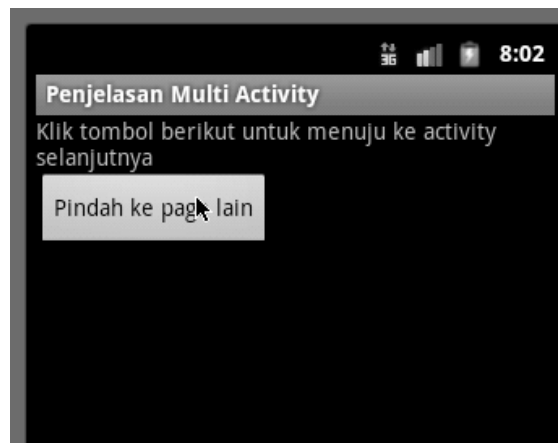
1. Buat proyek baru.
2. Kemudian ganti dulu app_name(String) di /res/values/string.xml dan ganti teks menjadi “Penjelasan Multi Activity”.

Gambar 1.68 Penggantian teks

3. Kemudian buka /res/layout/main.xml yang mendefinisikan tampilan dari “window” di Android. Gantilah isi main.xml ini dengan kode seperti berikut, yaitu berisi teks dan button yang bisa diklik untuk men-generate activity.

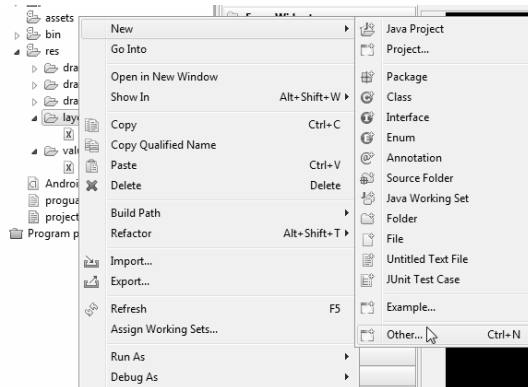
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Klik tombol berikut untuk
menuju ke activity selanjutnya"
            />
        <Button
                android:id="@+id/main_btnPage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Pindah ke page lain"
            />
</LinearLayout>
```

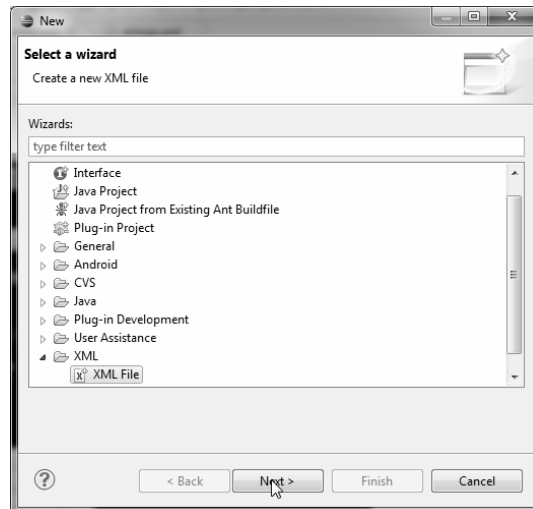
```
projek1 Manifest | main.xml | strings.xml

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Klik tombol berikut untuk menuju ke activity selanjutnya"
        />
    <Button
        android:id="@+id/main_btnPage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pindah ke page lain"
        />
</LinearLayout>
```

Gambar 1.69 Isi /res/layout/main.xml sudah diubah

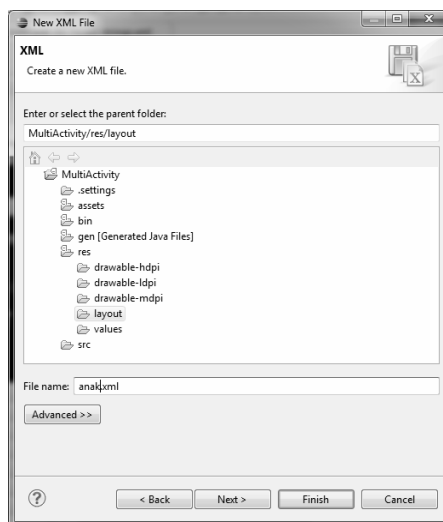
4. Kalau di-run dan dijalankan di ponsel/emulator, maka di tampilan muncul teks dan tombol yang bisa diklik.

Gambar 1.70 Tombol yang bisa diklik

5. Tapi ketika tombol diklik, belum ada apa-apanya. Karena Anda belum membuat intent untuk memproses pemindahan activity.

Gambar 1.71 Tampilan ketika tombol diklik

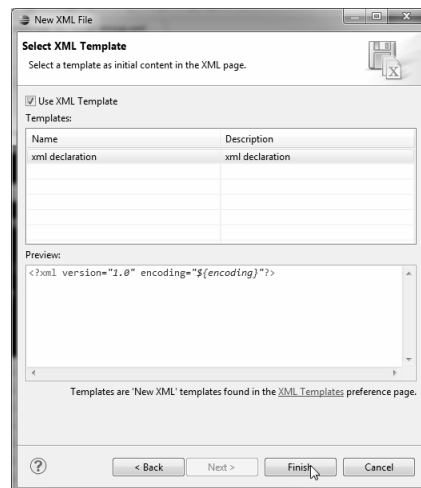
6. Sekarang Anda akan membuat tampilan layout baru lagi yang mirip dengan membuat “window” baru lagi di pemrograman VB.
7. Klik kanan pada folder /res/layout kemudian klik pada **New > Other**.

Gambar 1.72 Menu untuk membuat res/layout

8. Pilih **XML > XML File di Select a wizard**.

Gambar 1.73 Pemilihan XML File di select a wizard

9. Beri nama anak.xml yang menunjukkan bahwa window ini nanti akan menjadi anakan/child yang akan ditampilkan ketika user mengklik tombol di window pertama.

Gambar 1.74 Pemberian nama file xml di /res/layout

10. Kemudian pilih **Create XML from an xml template**.

Gambar 1.75 Create XML for an xml template

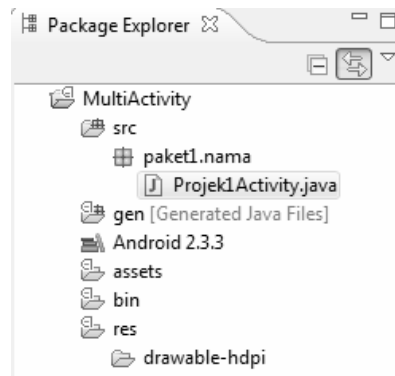
11. Kemudian klik **Finish** di tahap akhir.

Gambar 1.76 Klik finish untuk membuat file xml

12. Buka bagian **Source** dari file anak.xml ini.

Gambar 1.77 Klik pada Source di xml

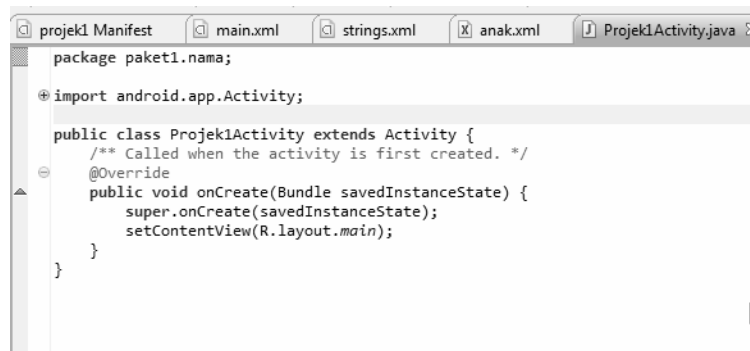
13. Kemudian isikan kode berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
        <TextView
               android:id="@+id/anak_txtPage"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Page lain yang diakses dengan
klik Button"
            />
        <Button
               android:id="@+id/anak_btnPage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Kembali lagi"
            />
</LinearLayout>
```

14. Kemudian klik pada file /src/nama_paket/nama_paket.java untuk mengedit kode di dalamnya.

Gambar 1.78 Pengeditan nama file activity

15. Kemudian lihat bagian import, secara default, yang diimpor hanya android.app.Activity saja.

projek1 Manifest | main.xml | strings.xml | anak.xml | Projek1Activity.java

```
package paket1.nama;

import android.app.Activity;

public class Projek1Activity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Gambar 1.79 Android.app.activity

16. Tambahkan import dengan baris-baris berikut:

```
import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
```

17. Hasilnya menjadi seperti berikut.

```
projek1 Manifest | main.xml | strings.xml | anak.xml
package paket1.nama;


import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.widget.Button;

public class Projek1Activity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

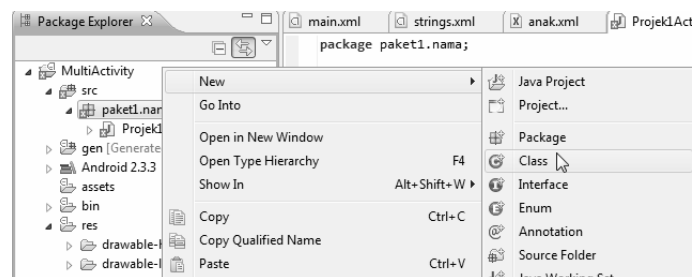
Gambar 1.80 Pengeditan file di class activity Projek1Activity

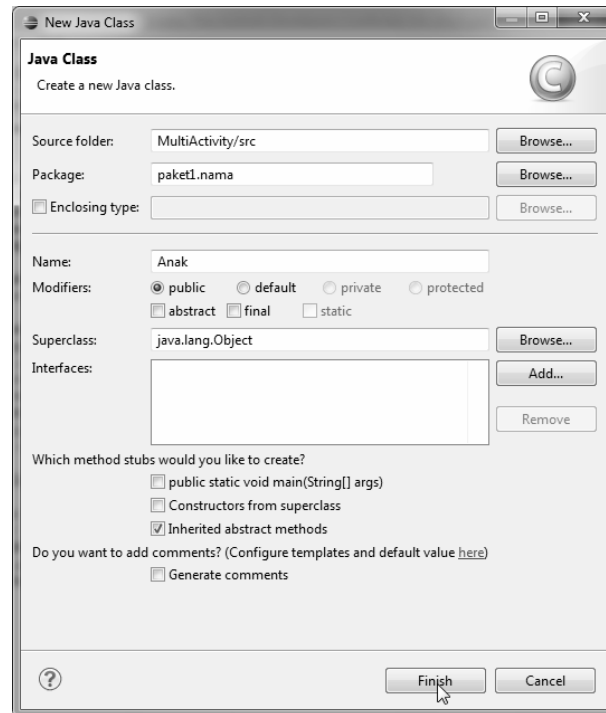
18. Kemudian untuk class default yang extends activity, Anda bisa mengeditnya seperti berikut (untuk kode yang lama diberi komentar dengan tanda //):

```
public class Projek1Activity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        //yg dikomenin ini kode asli
        //  super.onCreate(savedInstanceState);
        //  setContentView(R.layout.main);
         super.onCreate(savedInstanceState);
         setContentView(R.layout.main);
         //mendefinisikan button dengan nama 'go'.
         Button go =
(Button)findViewById(R.id.main_btnPage);
         //ketika button 'go' ini diklik,
         go.setOnClickListener(new View.OnClickListener()
{
           //maka berikut ini yang dilakukan
                       public void onClick(View v) {
                               // TODO Auto-generated
method stub
                       Intent i = new
Intent(Projek1Activity.this, Anak.class);
               i.putExtra("pesan", "dari activity
pertama");
                               startActivity(i);
                       }
               });

    }
}
```

19. Pada kode di atas, Anda mendefinisikan dulu button go di activity dengan mencari button bernama R.id.main.btnPage di layout. Setelah itu Anda mengeset go.Onclicklistener, yaitu apabila ada klik di button go ini, maka Anda mengatur action di public void onClick.

20. Di sini, Anda menentukan kode untuk berpindah, yaitu di intent. Di samping itu, ada tambahan kode, yaitu TextView diberi tambahan pesan.

21. Berikutnya, Anda bisa menentukan class activity untuk /res/layout/anak.xml yang belum digunakan. Klik **New** > **Class** di /src/paketjava.

Gambar 1.81 Klik menu untuk membuat class baru

22. Beri nama paket, misalnya Anak di name untuk class yang menampung activity.

Gambar 1.82 Pemberian nama untuk class yang akan meng-extend activity

23. Isikan class untuk menangani kode dari layout anak.xml ini dengan kode berikut:

```
public class Anak extends Activity {
	@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.anak);

        TextView txtChild =
(TextView)findViewById(R.id.anak_txtPage);
        Button go =
(Button)findViewById(R.id.anak_btnPage);

        //menerima Intent yang dikirim dari halaman utama
        Intent i = this.getIntent();
        txtChild.setText("Ini adalah halaman kedua
"+i.getStringExtra("pesan"));

        go.setOnClickListener(new View.OnClickListener()
{
```

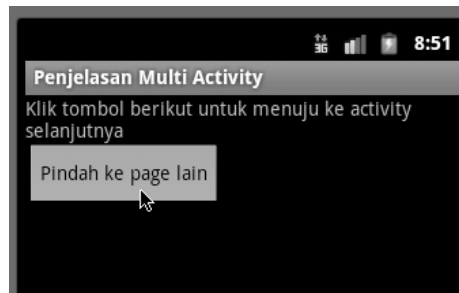
```
                        public void onClick(View v) {
                                // TODO Auto-generated
method stub
                                finish();
                        }
                });
        }
}
```

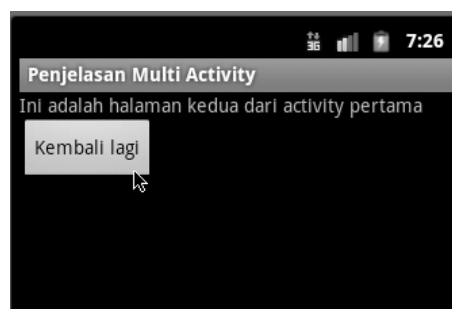
```
main.xml | strings.xml | anak.xml | Projek1Activity.java | MultiActivity Manifest | Anak.java
package paket1.nama;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class Anak extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.anak);

        TextView txtChild = (TextView)findViewById(R.id.anak_txtPage);
        Button go = (Button)findViewById(R.id.anak_btnPage);

        //menerima Intent yang dikirim dari halaman utama
        Intent i = this.getIntent();
        txtChild.setText("Ini adalah halaman kedua "+i.getStringExtra("pesan"));

        go.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                // TODO Auto-generated method stub
                finish();
            }
        });
    }
}
```

Gambar 1.83 Kode activity di dalam class untuk menangani anak.xml

24. Buka manifest kemudian tambahkan kode berikut untuk mengenalkan activity di class .Anak ke manifest.

```
        <activity android:name=".Anak"
                  android:label="@string/app_name">
            <intent-filter>
                <action
android:name="android.intent.action.VIEW" />
                <category
android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
```

25. Jika dijalankan, klik pada tombol **Pindah ke page lain**.

Gambar 1.84 Klik pada tombol

26. Maka intent akan bekerja dan memindahkan Anda ke window kedua, di mana Anda bisa melihat pesan dan pesan ini ditambahi text yang dibawa oleh intent. Jika tombol ini diklik lagi, muncul **Finish** untuk mengembalikan ke page pertama.

Gambar 1.85 Window kedua sudah muncul

1.5 Thread

Tiap aplikasi, secara default berjalan pada satu proses. Di mana proses ini akan menampung semua task. Agar antarmuka tidak hang, task yang sangat boros daya seperti download atau proses lainnya bisa diletakkan di thread berbeda di background, sesuai dengan keinginan programmer.

Tapi biasanya OS android ini akan memprioritaskan thread. Dan setiap aplikasi bisa memanfaatkan thread ini.

Jadi, memisahkan action ke sebuah thread adalah sebuah hack yang bisa Anda lakukan jika hendak melakukan sesuatu yang sangat penting. Misalnya yang berisiko untuk membuat android menjadi hang atau sangat mengonsumsi daya.

Berikut ini contoh kode menjalankan sebuah fungsi fiktif mainkan_audio() tanpa thread, alias menjalankan di activity seperti biasa:

```
Button startButton = (Button) findViewById(R.id.trigger);
startButton.setOnClickListener(new View.OnClickListener() {
public void onClick(View view){
// Contoh call pada fungsi yang memakan waktu
// fungsi ini memungkinkan thread utama menjadi hang
//fungsi ini hanya fiktif, jadi kalau dicoba
//belum terdaftar alias akan error.
    mainkan_audio();
}
});
```

Seandainya fungsi mainkan_audio() ini dieksekusi dan audio yang dimainkan sangat panjang, maka semua request untuk kembali ke home atau mengklik button tidak akan dilakukan sampai audio ini selesai dimainkan seluruhnya.

Dan kalau antarmuka user ini tidak responsif, maka bisa membuat sistem android menampilkan pesan error.

Gambar 1.86 Error pada sebuah activity karena sedang menjalankan hal yang sangat mengonsumsi daya atau waktu

Anda bisa mengakali atau melakukan hack untuk activity berat seperti ini dengan memisahkan action tersebut ke thread tersendiri. Dengan cara seperti berikut:

1. Buat sebuah thread untuk meng-hold objek yang bisa dijalankan. Thread ini akan terpisah dari activity utama. Sintaksnya seperti berikut:

```
Thread threadBackground = new Thread(
//objek yang mau di-run ditaruh di sini
);
```

2. Buat objek yang mau di-run menggunakan kode berikut, yaitu untuk menjalankan fungsi yang sangat mengonsumsi waktu:

```
new Jalankan() {
public void run() {
       mainkan_audio();
}
}
```

3. Anda bisa menjalankan thread, di mana thread ini menjalankan task menggunakan perintah berikut:

```
threadBackground.start();
```

1.5.1 Pengaturan Prioritas Thread

Sistem android meng-handel prioritas thread. Secara default, thread baru memiliki prioritas 5. Anda bisa mengatur priortias dari thread ini dengan fungsi setPriority seperti berikut:

```
threadKu.setPriority(priority)
```

Pengesetan prioritas ini sebelum thread dijalankan:

```
threadKu.start().
```

Pengisian prioritas tidak bisa lebih besar dari Thread.MAX_PRIORITY (yaitu 10) atau lebih rendah dari Thread.MIN_PRIORITY (yaitu 1).

1.5.2 Membatalkan Thread

Ketika ada komponen yang selesai atau dimatikan, maka thread juga biasanya harus dimatikan. Contohnya thread:

```
private volatile Thread threadKu;
```

Method threadKu.stop() sudah deprecated karena sudah meninggalkan aplikasi pada kondisi yang tidak bisa diprediksi. Sebagai gantinya, Anda bisa menggunakan contoh berikut untuk melakukan penghentian thread.

```
//gunakan untuk men-stop thread threadKu
if(threadKu != null) {
Thread dummy = threadKu;
threadKu = null;
dummy.interrupt();
}
```

Pada level aplikasi, ada banyak cara untuk melakukan ini. Yaitu dengan mengeset thread menjadi daemon.

```
//Gunakan ketika awal memulai thread
threadKu.setDaemon(true);
threadKu.start();
```

1.5.3 Contoh Menggunakan Thread dalam Aplikasi

Berikut ini contoh penggunaan thread dalam aplikasi. Thread di sini digunakan untuk me-run action yang melibatkan perhitungan yang cukup susah, di mana perhitungan ini kalau dijalankan melalui activity biasa, bisa membuat android jadi berat.

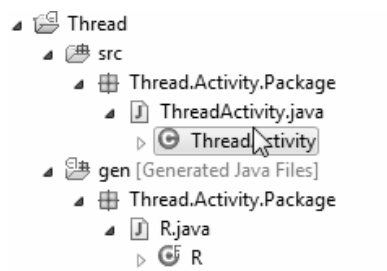
1. Buka /res/layout/main.xml dan beri kode seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/trigger"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Silakan klik" />

 <TextView
               android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Teks Awal"
 />
</LinearLayout>
```

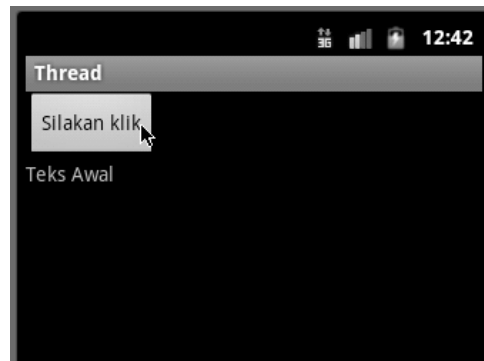
Gambar 1.87 Kode /res/layout/main.xml yang menampilkan button dan textview

2. Kode di atas akan membuat adanya button dan textview, masing-masing bernama “trigger” dan “text”.

3. Sekarang buka activity untuk proyek thread ini di folder /src/nama_paket.

Gambar 1.88 Membuka file .java untuk mengatur activity

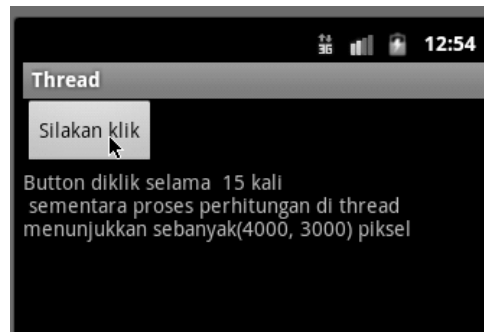
4. Isikan kode berikut di class activity untuk thread Anda:

```
package Thread.Activity.Package;

import android.app.Activity;
public class ThreadActivity extends Activity implements
Runnable {
int jmlKlik=0;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
final TextView tv  = (TextView) findViewById(R.id.text);
/* deteksiPiksel(); */
//membuat thread background agar task yang boros
dijalankan via thread
Thread thread = new Thread(ThreadActivity.this);
thread.start();
Button startButton = (Button) findViewById(R.id.trigger);
startButton.setOnClickListener(new View.OnClickListener()
{
public void onClick(View view){
tv.setText("Button diklik selama  " + ++jmlKlik
+ " kali \n sementara proses perhitugnan di thread
menunjukkan sebanyak"
+ "(" + xi + ", " + yi + ") piksel");
}
});
}
public void run() {
deteksiPiksel();
}
//Pendeteksian piksel
int xi, yi;
private double deteksiPiksel() {
int x_pixels = 4000;
int y_pixels = 3000;
double transformasi_gambar=0;
//loop ganda untuk membuat proses menjadi lama dan berat
//ditambah pemrosesan lain yang lebih panjang seperti
cosinus
//projek ini tidak ada tujuannya, hanya untuk mendemokan
pemrosesan yang lama
//sehingga jika tidak dijalankan via thread bisa hang
android-nya
for(xi=0; xi<x_pixels; xi++) {
for(yi=0; yi<y_pixels; yi++) {
transformasi_gambar = Math.cosh(xi*yi/x_pixels/y_pixels);
}
}
return transformasi_gambar;
}
}
```

5. Jalankan program. Klik pada **tombol**.

Gambar 1.89 Klik pada tombol untuk mengaktifkan thread

6. Hasilnya, di bawah muncul teks dan perhitungan yang terus berlangsung menggunakan thread. Ketika perhitungan ini berlangsung, Anda masih bisa klik button, artinya UI (user interface) masih responsif. Ini adalah salah satu fitur thread.

Gambar 1.90 Perhitungan berlangsung

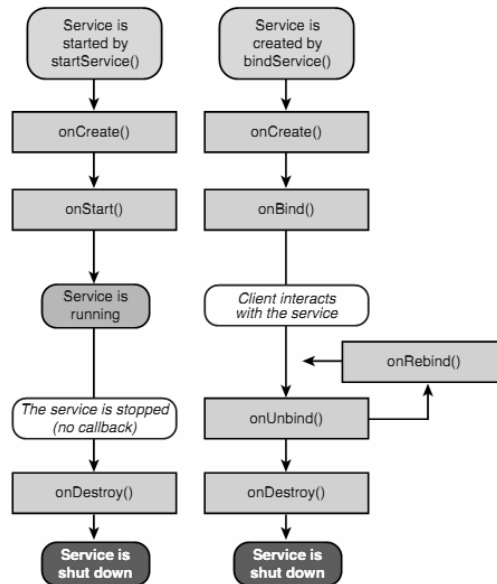
7. Klik lagi, maka angka langsung berubah lagi. Ini artinya thread masih bekerja.

Gambar 1.91 Thread masih bekerja

1.6 Service

Service merupakan komponen android yang berjalan di backgruond tanpa interaksi dari user. Service bisa di-start dan di-stop oleh komponen.

Ketika sedang running, komponen juga bisa berhubungan/bind ke service ini. Service juga bisa men-stop dirinya sendiri. Berikut ini contoh service:

1. Sebuah activity memungkinkan user memilih file audio, kemudian men-start service yang memainkan file. Selama memainkan, ada activity baru yang mem-bind ke service yang sedang berjalan yang memungkinkan user untuk mengganti file audio atau men-stop pemutaran file audio tersebut.
2. Activity memulai service untuk meng-upload gambar ke website. Dan ada activity baru yang mem-bind ke service untuk menentukan file yang sedang di-upload dan menampilkan gambarnya ke layar.

Berikut ini contoh siklus hidup sebuah service.

Gambar 1.92 Siklus hidup produk

1.6.1 Membuat Self-Contained Service

Langkah-langkah untuk membuat service yang mandiri dan diasosiasikan dengan komponen tunggal adalah:

1. Buatlah sebuah class yang meng-extend service, di Eclipse, klik kanan dan pilih **New > Class** lalu pilih Android.app.
2. Deklarasikan layanan di AndroidManifest.xml dengan menambahkan variasi tag berikut.

   ```
   <service android:name=".myService"></service>
   ```

3. Gunakan method onCreate() dan onDestroy().
4. Gunakan method onBind() jika ada komponen baru yang mem-bind ke ini.

5. Aktifkan service dari pemicu eksternal, ini karena service tidak bisa jalan sendiri, tapi butuh pemicu dari luar. Misalnya komponen lain bisa meng-create intent untuk menjalankan dan menyetop service menggunakan fungsi startService() atau stopService().

Nantinya Anda bisa menerapkan konsep service, thread, dan lainnya ini lebih lanjut ketika sudah memahami konsep multimedia di Android dan konsep-konsep advanced lainnya.

1.7 Broadcast Receiver

Broadcast receiver bertugas mendengarkan pesan broadcast penting untuk men-trigger sebuah event. Di antara contoh penggunaan broad-cast receiver adalah:

- Tombol di kamera diklik
- Baterai low
- Ada aplikasi baru yang terinstal

Komponen yang di-generate oleh user bisa juga mengirimkan broadcast. Misalnya:

- Kalkulasi sudah selesai
- Thread tertentu sudah dimulai

Class broadcast receivers meng-extend class BroadcastReceiver atau salah satu dari sub-class-nya. Siklus hidup dari broadcast lifecycle sangat sederhana.

Method onReceive() dipanggil ketika ada pesan ke receiver. Setelah method ini selesai, instance BroadcastReceiver menjadi tidak aktif.

Broadcast receiver biasanya memulai komponen terpisah atau mengirmi notifikasi ke user melalui method OnReceive().

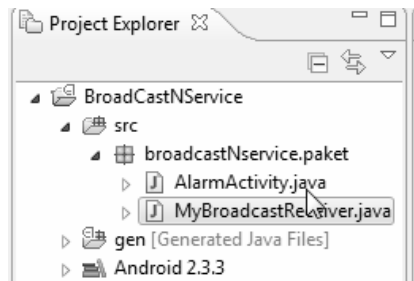
Jika broadcast receiver hendak melakukan sesuatu yang lebih memakan waktu, biasanya menggunakan service dan bukannya menelurkan thread karena boradcast receiver yang tidak aktif biasanya dimatikan oleh sistem operasi.

Untuk lebih jelasnya, Anda bisa mengikuti contoh program berikut yang mengimplementasikan service dan broadcast receiver:

1. Buat proyek baru, misalnya BroadcastNService.

2. Kemudian buka /res/layout/main.xml dan masukkan editText dan button masing-masing dengan id = “waktu” dan id = “hitung”. Atau isi /res/layout/main.xml menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
        android:orientation="vertical"
android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <EditText android:layout_height="wrap_content"
android:id="@+id/waktu"
                android:layout_width="wrap_content"
android:hint="Jumlah detikan"

        android:inputType="numberDecimal">></EditText>
        <Button android:text="Mulai hitung"
android:id="@+id/hitung"
                android:onClick="startAlert"
android:layout_width="wrap_content"

        android:layout_height="wrap_content"></Button>
</LinearLayout>
```

3. Kemudian buat file class untuk menuliskan Broadcast Receiver. Misalnya nama file class-nya MyBroadcastReceiver.

Gambar 1.93 MyBroadcastReceiver

4. Beri kode untuk menyalakan vibrator, di atasnya ada beberapa import untuk menampilkan pesan dalam bentuk toast dan mengaktifkan vibrator.

```
package broadcastNservice.paket;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Vibrator;
import android.widget.Toast;

public class MyBroadcastReceiver extends
BroadcastReceiver {
        @Override
        //kode berikut untuk menjalankan alarm memakai
broadcast receiver
        public void onReceive(Context context, Intent
intent) {
                Toast.makeText(context, "Alarm menyala
sekarang!!!!.",
                                Toast.LENGTH_LONG).show();
                // menggetarkan ponsel
                Vibrator vibrator = (Vibrator)
context.getSystemService(Context.VIBRATOR_SERVICE);
                vibrator.vibrate(3000);
        }

}
```

5. Edit android manifest untuk mendefinisikan <receiver> dan mengacukannya ke broadcast receiver yang barusan dibuat. Set juga user-permissions untuk mengizinkan vibrasi/getaran.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android
"
```

```
    package="broadcastNservice.paket"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".AlarmActivity" >
            <intent-filter >
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver
android:name="MyBroadcastReceiver"></receiver>
    </application>
       <uses-permission
android:name="android.permission.VIBRATE"></uses-
permission>
</manifest>
```

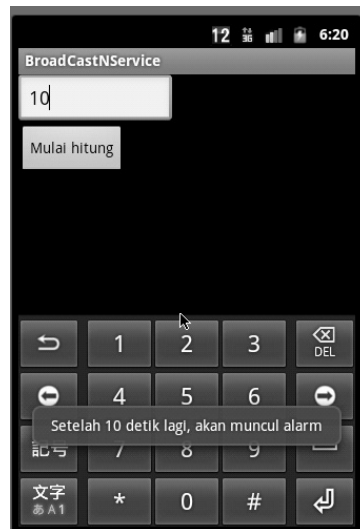
6. Kemudian edit activity untuk memberi kode untuk mengambil nilai detikan dan kemudian mem-broadcast dan memanggil.

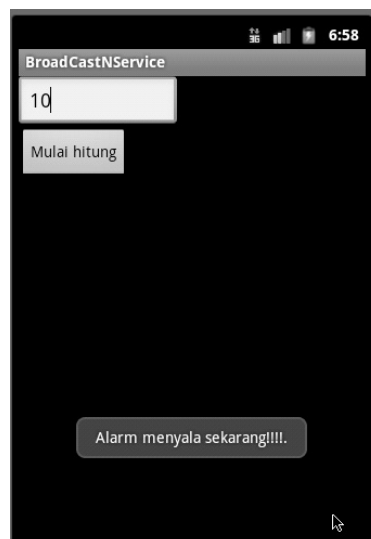
```
package broadcastNservice.paket;

import android.app.Activity;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class AlarmActivity extends Activity {
    /** Dipanggil ketika activity di-buat pertama kali.
*/
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        //me-load layout
    }
    public void startAlert(View view) {
       //memproses ketika button diklik, maka nilai
detikan diambil
               EditText text = (EditText)
findViewById(R.id.waktu);
               //nilai detikan ini diubah ke integer
```

```
                int i =
Integer.parseInt(text.getText().toString());
                Intent intent = new Intent(this,
MyBroadcastReceiver.class);
                //dipindahkan ke service dan broadcast
manager
                PendingIntent pendingIntent =
PendingIntent.getBroadcast(

        this.getApplicationContext(), 1, intent, 0);
                AlarmManager alarmManager = (AlarmManager)
getSystemService(ALARM_SERVICE);
                alarmManager.set(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis()
                               + (i * 1000),
pendingIntent);
                Toast.makeText(this, "Setelah " + i + "
detik lagi, akan muncul alarm",
                               Toast.LENGTH_LONG).show();
        }
}
```

7. Kemudian run di emulator (atau kalau mau merasakan efek vibrator, Anda bisa menjalankan di ponsel fisik). Isikan detikan untuk menyalakan alarm, kemudian klik tombol.

Gambar 1.94 Pengisian nilai untuk broadcast receiver

8. Muncul toast ketika button diklik.

Gambar 1.95 Muncul toast yang menyatakan bahwa muncul alarm setelah beberapa waktu

9. Kalau alarm sudah menyala, muncul toast disertai dengan vibrasi. Vibrasi ini diaktifkan oleh broadcast receiver.

Gambar 1.96 Alarm sudah menyala